

AL-TR-1993-0050

AD-A266 951



RAMCAD ADVANCED RESEARCH FINAL REPORT

Matthew C. Tracy, II

HUMAN RESOURCES DIRECTORATE
LOGISTICS RESEARCH DIVISION

Charles T. Kitzmiller
Michael W. Anderson

BOEING COMPUTER SERVICES
RESEARCH AND TECHNOLOGY
P.O. BOX 24346, M.S. 7L-64
SEATTLE, WASHINGTON 98124-0346

APRIL 1993

DTIC
ELECTE
JUL 22 1993
S B D

FINAL TECHNICAL REPORT FOR PERIOD SEPTEMBER 1987 - MARCH 1993

Approved for public release; distribution is unlimited.

93 7 21 00 4

93-16500



9301

AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-8573

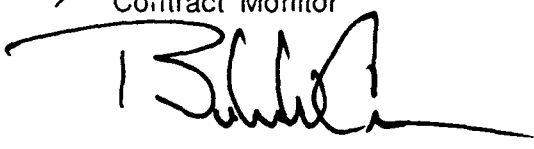
NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation, or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed and is approved for publication.



MATTHEW C. TRACY, II
Contract Monitor



BERTRAM W. CREAM, Chief
Logistics Research Division

REPORT DOCUMENTATION PAGEForm Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE April 1993	3. REPORT TYPE AND DATES COVERED Final - Sep 1987 - Mar 1993	
4. TITLE AND SUBTITLE RAMCAD Advanced Research Final Report			5. FUNDING NUMBERS C - F33615-87-C-0001 PE - 63106F PR - 2940 TA - 01 WU - 04	
6. AUTHOR(S) Matthew C. Tracy, II Michael W. Anderson Charles T. Kitzmiller				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Armstrong Laboratory Human Resources Directorate Logistics Research Division WPAFB, OH 45433-6503 Boeing Computer Services Research and Technology P.O. 24346, M.S. 7L-64 Seattle, WA 98124-0346			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Armstrong Laboratory Human Resources Directorate Logistics Research Division Wright-Patterson AFB, OH 45433-6573			10. SPONSORING/MONITORING AGENCY REPORT NUMBER AL-TR-1993-0050	
11. SUPPLEMENTARY NOTES Armstrong Laboratory Technical Monitor: Matthew C. Tracy, AL/HRGA, (513) 255-8399				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The goal of the Reliability, Availability, and Maintainability in Computer-Aided Design (RAMCAD) research effort is to define a design environment that fully supports reliability, maintainability, and supportability (RM&S) issues through the use of computer-aided design/computer-aided engineering (CAD/CAE) workstations. The goal of the RAMCAD advanced research effort was to define a computer-aided methodology to help design engineers optimize designs relative to RM&S issues. This report is the final report for the contractual effort and describes the approach taken during the contract, the results of the contract, and where this research could continue in the future.				
14. SUBJECT TERMS computer-aided design computer-aided engineering		design maintainability methodology optimization	RAMCAD reliability	15. NUMBER OF PAGES 96
17. SECURITY CLASSIFICATION OF REPORT Unclassified		18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	16. PRICE CODE
				20. LIMITATION OF ABSTRACT SAR

CONTENTS

	Page
FIGURES	vi
TABLES	vi
PREFACE	vii
I. INTRODUCTION	1
Scope	1
Approach	1
Report Organization	3
II. DESIGN AND METHODOLOGY RESEARCH	3
Developing a Research Plan	4
Analyzing the Design Process	5
Analyzing Current Computer-Aided Design/Computer-Aided Engineering Capabilities	8
Specifying the Reliability, Maintainability, and Supportability Design and Analysis Needs	9
Reliability, Maintainability, and Supportability Design Practices	10
Reliability, Maintainability, and Supportability Technology Needs	11
Specific Research Area	14
Developing a Casebook	16
Developing Adjudication and Sequencing Rules	16
Developing the Design Methodology	17
III. SOFTWARE TOOL CODING	22
Developing the System for the Interactive Design and Computer Analysis of Reliability	23
Exploration of Parts	25
Exploration of Temperature Changes	26
Exploration of Marginal Benefits	26
Exploration of Combinational Changes	27
Developing the Inherent Testability Analyzer	27
Define Controllability and Observability	29
Controllability Propagation	29
Observability Propagation	30
Component Testability Analysis	30
Developing the Statistical Testability Analyzer	30
Input Hardware Test Characteristics	31
Input Test Coverage Data	31
Run Statistical Testability Analyzer	31
Generate Statistical Testability Analyzer Reports	32
Make Necessary Design and Test Modifications	32
Developing Scan Assistant	32

IV. EVALUATION AND VALIDATION	33
Developing Generic Evaluation Criteria	34
Product Quality	34
Process Cost	36
Process Time.....	37
Process Flexibility.....	39
Job Satisfaction	41
Evaluation Challenges and Approaches	42
Evaluation Challenges	42
Evaluation Approaches	44
Developing Evaluation Criteria and Methods and Evaluating and Validating the Research.....	46
System for the Interactive Design and Computer Analysis of Reliability and Statistical Testability Analyzer Metrics and Evaluation.....	47
Inherent Testability Analyzer Evaluation Metrics and Evaluation.....	53
General Discussion and Conclusion	56
V. LESSONS LEARNED	57
Optimization.....	57
Design Data, Knowledge, and Expertise	59
Similarity Analysis/Evaluation.....	60
Evaluator Tools By Themselves Are Not Sufficient	61
Rules, Heuristics, and Guidelines Rationale.....	62
Computer Support for Design	62
Design Representation	64
VI. FUTURE RESEARCH.....	65
General Research Areas	65
Design Process, Methodology, and Knowledge.....	65
Design Data and Libraries	66
Design Estimates and Predictions	67
Design Monitoring and Advice	68
Reliability.....	69
Reliability Prediction and Modeling	69
Failure Modes Analysis.....	70
Reliability Allocation	70
Reliability Enhancement	71
Maintainability	71
Maintainability Prediction and Modeling	72
Maintainability Allocation	72
Maintainability Enhancement.....	72
Testability Design Aids	73
Human Factors	73
Supportability.....	74
Key Supportability Characteristics Prediction and Modeling.....	74

FIGURES

Figure	Page
1 Three Phases of the Research Effort	2
2 Phase 1 Schedule	4
3 Research Plan Basic Design Cycle	6
4 Competing Design Requirements	7
5 Current Detailed Design Process.....	18
6 Proposed Detailed Design Process	21
7 Phase 2 Schedule	23
8 System for the Interactive Design and Computer Analysis of Reliability Sample Output for the Exploration of Combinational Changes.....	28
9 Phase 3 Schedule	34

TABLES

Table	Page
1 Capabilities Desired in a Computer-Aided Design/Computer-Aided Engineering Environment.....	9
2 Reliability Enhancement Techniques	24

PREFACE

In the early 1980s, the Department of Defense created programs such as Reliability and Maintainability (R&M) 2000 to highlight the importance of improving weapon system designs. Consequently, the Armstrong Laboratory, Logistics Research Division (AL/HRG) initiated the Reliability, Availability, and Maintainability in Computer-Aided Design (RAMCAD) Program. The overall goal of the RAMCAD Program is to create a design environment that assists design engineers effectively consider reliability, maintainability, and supportability issues through the use of computer-aided design/computer-aided engineering (CAD/CAE) workstations, thus improving weapon systems.

One aspect of the RAMCAD Program involves performing long-term research that would lead to the creation of a design optimization methodology and proof-of-concept software tools that implement aspects of the methodology for a CAD/CAE workstation. In September 1987, AL/HRG awarded Boeing Computer Services (BCS) a contract to perform this research. The BCS effort focused on the development of a methodology and a set of methods and techniques that design engineers could employ to optimize the design of an electronic system within given reliability, testability, performance, cost, area, and other requirements.

This report summarizes the research performed by BCS under contract F33615-87-C-0001. It describes the technical approach of the research effort, presents the findings of the research, and identifies areas that should be expanded in the future.

The authors would like to thank all of the people who contributed to both this report and the associated research effort. In particular, thanks go to the following people: Alex Bobotek, Peter Russo, and Judy Powell of BCS for all their tireless work on this effort; Chuck Yount and Inderpal Bhandari of Carnegie-Mellon University (CMU) who, with support of Dr. Dan Siewiorek, put numerous hours into developing and modifying the System for the Interactive Design and Computer Analysis of Reliability (SIDEAR) and Scan Assistant (ScanIt); Bill Birmingham who worked on this effort both at CMU and at the University of Michigan; and finally Jerry Hollingsworth of Boeing, who provided us with his insights on the realities of the design process.

I. INTRODUCTION

This report summarizes the research performed by Boeing Computer Services (BCS) in support of the Armstrong Laboratory, Logistics Research Division (AL/HRG) Reliability, Availability, and Maintainability in Computer-Aided Design (RAMCAD) Software Development Program under contract F33615-87-C-0001. The purpose of this RAMCAD Program contracted effort is to develop methods and techniques to aid design engineers in creating better designs by improving access to existing design knowledge, rules, and guidelines throughout the design cycle.

Scope

This report describes the work accomplished by BCS during a 36-month technical effort aimed specifically at creating an enhanced approach to electronic system design that uses intelligent design tools to help design engineers¹ optimize a design for performance, cost, reliability, maintainability, and other requirements. In particular, the focus of this effort was to research and develop methods and tools that design engineers could employ to perform the trade-off analyses needed to apportion and allocate design requirements and goals, and hardware resources (e.g., area, power, and costs) from the system level, through the subsystem levels, to the detailed design level.² This effort also included methods and tools that design engineers could use to determine how well a proposed design meets its design requirements and to verify that the design does not exceed the assigned resources at any level in the design hierarchy.

Approach

To create an understandable and workable technical effort, this research effort was divided into three phases (see Figure 1). The first phase included the up-front research needed to create a possible answer to the question, "Where can intelligent design tools provide the most help to design engineers, both today and in the future, and what tasks must these tools perform?" The

¹ To avoid confusion, three terms will be used to define people who work on designs. The first term, system engineer, denotes individuals working on system-level problems that include such tasks as allocating resources and requirements to subsystems. The second term, detailed designer, refers to those who design hardware implementations that perform the functions required of one section of a subsystem (e.g., one printed circuit board) as determined by system engineers. The third term, design engineers, includes both system engineers and detailed designers.

² For modern electronic systems, reliability and testability in one form or another (e.g., built-in fault detection, built-in-test, built-in-test-equipment, on-board maintenance equipment, and external test equipment) drive maintenance and support costs. Consequently, these two factors became the dominant factors in this research effort.

answer to this question had to address whether a new design methodology needed to be proposed and identify the software tools needed to enable design engineers to employ this methodology.

The second phase involved creating proof-of-concept software tools to:

- demonstrate elements of the proposed methodology in order to solicit feedback from engineers,
- provide a method of evaluating and validating the methodology and the proposed design methods and techniques, and
- provide an understanding of the issues and problems involved in implementing a RAMCAD design system.

The third phase involved creating evaluation metrics and methods to be used in evaluating design tools and validating design methodologies in general as well as the tools and methodologies created specifically as part of this research effort. An evaluation and validation were then performed to determine the utility of the results of the first two phases. In addition, the evaluation and validation were used to solicit engineer feedback on the proposed approach and the continuation of this research area. Each phase is discussed in depth as part of this report.

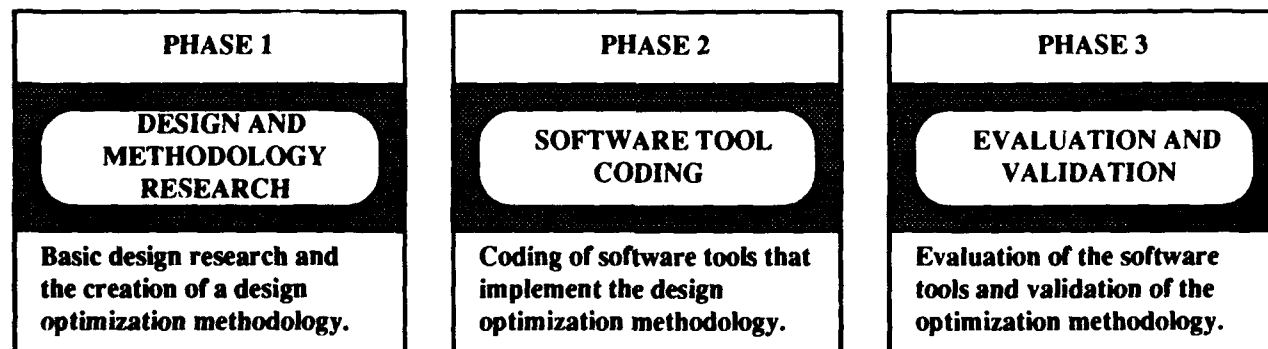


Figure 1
Three Phases of the Research Effort

Report Organization

The remainder of this report is organized into five sections as follows.

- a. Section II discusses the basic design and methodology research performed in Phase 1.
- b. Section III addresses the coding of the software tools performed in Phase 2.
- c. Section IV discusses the evaluation and validation performed in Phase 3.
- d. Section V describes the lessons learned during this research effort.
- e. Section VI suggests future research directions for this type of effort as well as other areas that should be pursued.

II. DESIGN AND METHODOLOGY RESEARCH

The objective of Phase 1 of the research effort was to develop a methodology that would help answer the question: "Where can intelligent design tools provide the most help to design engineers, both today and in the future, and what tasks must these tools perform?" This phase included seven separate but interrelated tasks: (1) developing a research plan; (2) analyzing the design process; (3) analyzing the current state of computer-aided design/computer-aided engineering (CAD/CAE) capabilities; (4) specifying reliability, maintainability, and supportability (RM&S) design and analysis needs; (5) developing a casebook of RM&S design and analysis rules, heuristics, and guidelines (RH&Gs); (6) developing adjudication and sequencing rules for the casebook RH&Gs; and (7) developing an RM&S design methodology. The Phase 1 schedule is shown in Figure 2. Each Phase 1 task is discussed in greater detail in the remainder of this section.

The approach used for Phase 1 was to form a RAMCAD working group of experienced design engineers, RM&S specialists, and design researchers. The working group included personnel from each major Boeing division, Carnegie-Mellon University (CMU), The Ohio State University, and the University of Michigan. Within this group, subgroups were created to work on different aspects of the research problem and report their progress to the working group. Each subgroup was tasked with developing a working hypothesis, draft document, or prototype software. The research began with brainstorming sessions to determine possible solutions and sources of information, followed by a review of Department of Defense (DoD) and Boeing standards and handbooks as well as any pertinent documents and reports found through searches of available

document databases (e.g., Defense Technical Information Center). Finally, the subgroup performed interviews with design engineers from the Boeing design community. The design engineers were selected based upon the recommendations of their peers and the results of an initial screening interview. Special care was taken to ensure that the interviewees included those individuals recognized by the Boeing design community as particularly knowledgeable in the specific research problem being addressed by the subgroup. Next, the subgroup compiled the results and presented them to the working group for review and revision. The hypothesis or draft document of the subgroup was subsequently modified, as needed, to accommodate the views of the working group majority. The working group then accepted the results as its official position.

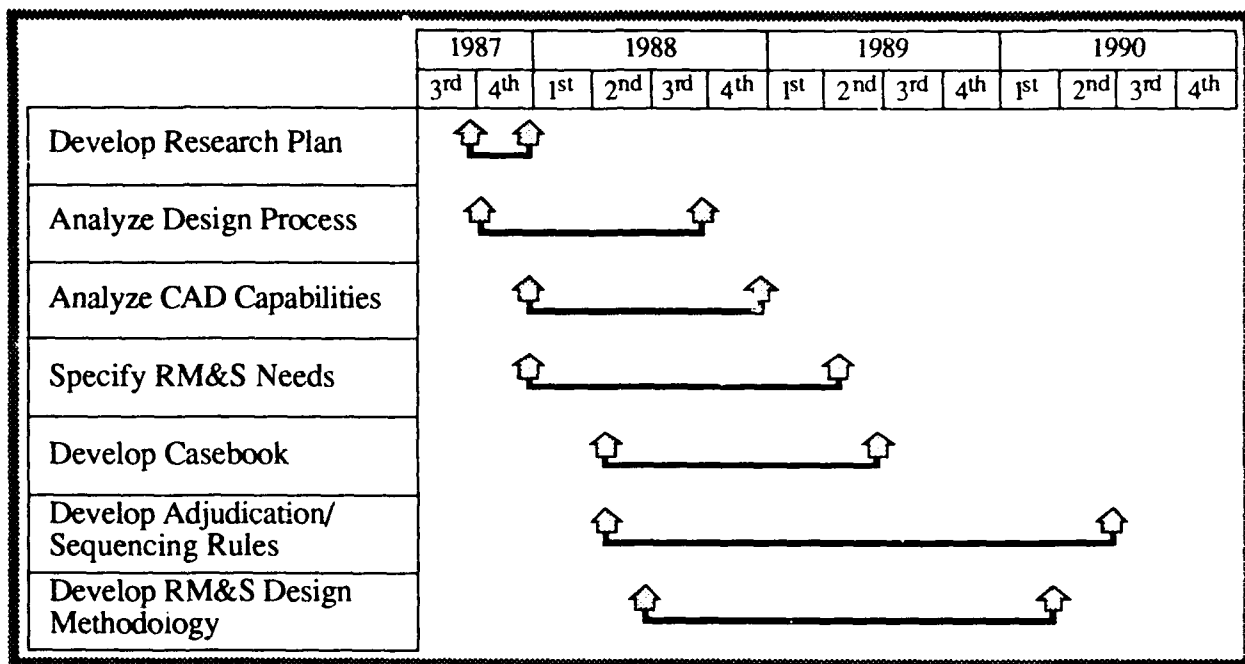


Figure 2
Phase 1 Schedule

Developing a Research Plan

A research plan was developed at the beginning of this effort to define the basic assumptions and goals of the research and the technical approach that was employed to perform the research. The root assumption was that, although RM&S considerations should permeate the design process to support system operational availability and minimize life-cycle cost (LCC), in practice they are usually addressed only during the later stages of the design cycle (shown in Figure 3³) for a variety

³ Both Figure 3 and Figure 4 are taken directly from the RAMCAD Software Development Program Research Plan [Rev. 2] (BCS, 1990a).

of reasons (see Kitzmiller & Anderson, 1991). The most significant reasons are constrained time and cost budgets, and insufficient access to information normally obtained through specialists. Although the thrusts in total quality management and integrated product development (IPD)⁴ have allowed many design engineers more freedom to consider RM&S earlier in the design cycle, information access is still a large problem.

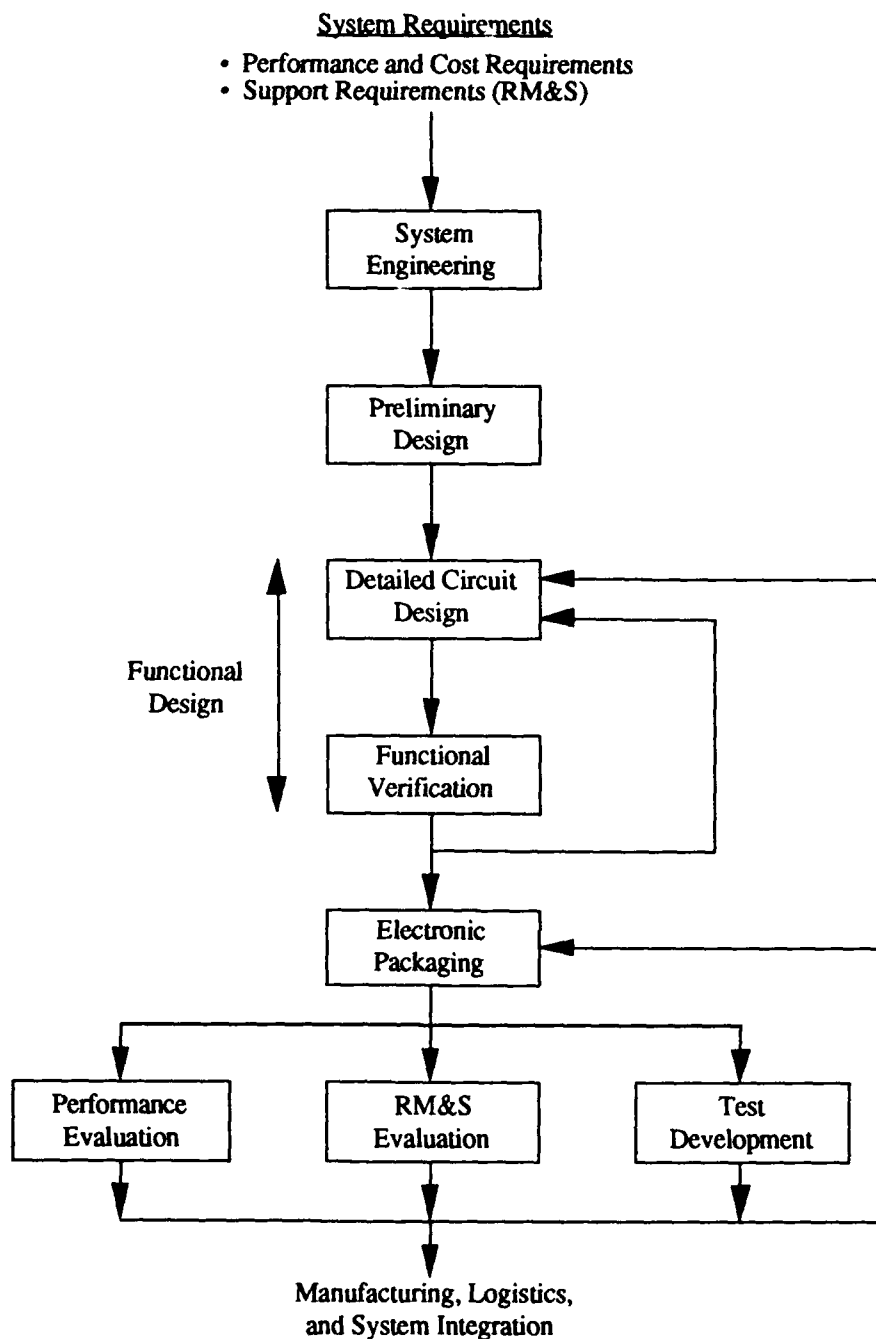
Because of the breadth of electronic design and the size of the research objective, it was determined that this effort should focus solely on avionics design. However, a concerted attempt was made to ensure that the methodology created under this effort was broad enough to eventually include other aspects of electronic design and, possibly, mechanical and structural design as well. In addition, avionics design offered both a reduced scope and plenty of competing design requirements (some of the competing design requirements are shown in Figure 4). Finally, it was agreed that any new methodology should help an engineer not only meet RM&S design criteria and goals, but assist in the designing of electronic systems to enhance the RM&S aspects of the design without unduly penalizing the performance and functionality of the system.

The research plan also served as the focus document for the research effort and included the schedules and goals for all three phases of the effort. Through the plan, each person could be reminded of the basic assumptions used throughout the effort and determine how any individual efforts supported the final goal.

Analyzing the Design Process

The first step to providing a good technical base for the research was the development of a detailed description and model of the current electronic design process to facilitate an analysis of current RM&S practices. The purpose of this task was to develop the information and understanding needed to identify specific problems in the design process, design methods, and associated analytical methods. It was originally assumed that a large percentage of the design process data needed would be documented in one or more reports. However, a very thorough literature search proved this assumption false. Although there are many reports that discuss the high-level tasks and concepts behind the design process, and work is continually being performed to improve design tools and knowledge, there was insufficient documented research that detailed the process at a low enough level to identify the best research areas for this effort. Furthermore, the documented research described how design engineers should perform the process rather than

⁴ IPD is also called concurrent engineering and simultaneous engineering.



Notes. Not all feedback loops are shown. This figure represents the basic design cycle in 1987. The current practice is now different due to an increasing emphasis on RM&S and such policies as TQM and IPD.

Figure 3
Research Plan Basic Design Cycle

the approach actually used on a daily basis. The documented research also described items common to most companies; it did not include company-specific tasks. However, the working group believed that this undocumented information might be a vital part of the design process; therefore, it might be critical to understanding how the low-level details of the design process interact and the location of design process shortcomings. Consequently, the analysis and evaluation of the design process became a much bigger effort than originally envisioned.

The results of this part of the effort are documented in a technical paper published by AL/HRG (Kitzmilller & Anderson, 1991). The paper provides an overview of the design process and in-depth discussions on both the systems engineering and detailed design tasks associated with electronics design. It also provides a brief discussion of some factors the design engineer must consider during each aspect of the design cycle and states some of the main problems of the design process.

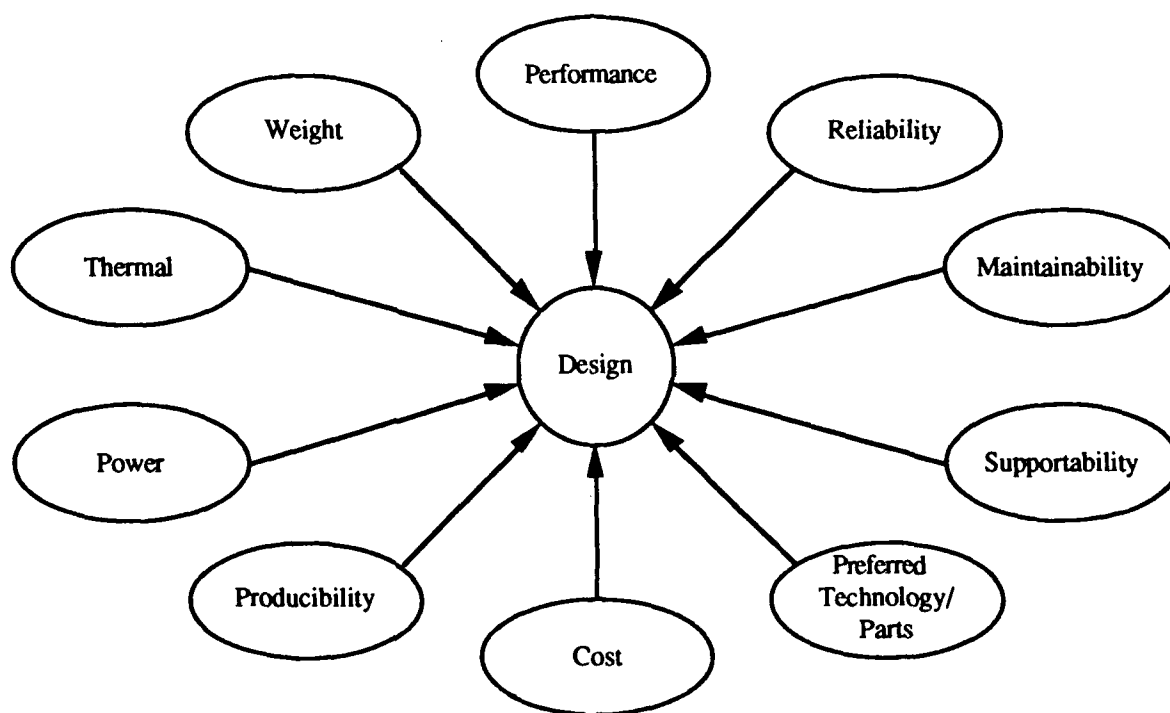


Figure 4
Competing Design Requirements

Analyzing Current Computer-Aided Design/ Computer-Aided Engineering Capabilities

The second step to providing a good technical base for the research effort was to assess current and near-term CAD/CAE capabilities. Because of the speed at which CAD/CAE systems have been evolving and the fact that most reports on the tools describe the capabilities of specific tools and not the shortfalls of a tool area (e.g., schematic capture and detailed design testability), past reviews and reports of software capabilities could not be relied upon. Instead, this aspect of the research needed to be accomplished with the aid of the Boeing design automation community.

One significant finding of this research was that design engineers accomplish much of the design work with pencil and paper. They typically use automated tools to document and verify the product of the design effort rather than as an environment in which to perform the design effort. This provides an obvious and major barrier to achieving the RAMCAD Program goal of aiding the design engineer through better automated tools. Any help offered by improved design tools would likely be provided too late in the design cycle to be entirely effective. Interviews with the design engineers provided a list of capabilities needed in future design environments to ensure the design engineers could use the tools to perform the design effort instead of only documenting it. This list, presented (in part) in Table 1, is not all inclusive; it is more a list of items the design engineers believe are missing in current environments than a reasoned specification for the ideal environment. This list does not include many obvious requirements such as schematic capture. The items in this list are not presented in any particular order and range from very broad applications to single, specialized activities.

As part of the effort to identify the characteristics of a future design environment, interviewees were also questioned about the attributes needed or desired in a future electronic CAD/CAE (ECAD/ECAE) user interface. This proved to be a poor way to define a user interface because most design engineers are not knowledgeable of user-interface technology and are not aware of the sophisticated capabilities being created for current and future user interfaces. The interviewees tended to describe features they have seen and liked versus imagining what was possible and most desirable. Furthermore, although design engineers should be the final judges of the usability of any CAD/CAE interface, they are likely to be less effective in defining it than would experts in the field.

The results of this research are documented in an interim report that describes the major functions of a knowledge-based design environment (BCS, 1988).

**Table 1. Capabilities Desired in a Computer-Aided Design/
Computer-Aided Engineering Environment**

- Access to all tools in a single environment without explicit data reformatting.
- Flexibility to use the tools in a manner that corresponds to the user's view of the task.
- Creation of objects with minimal detail and the ability to fill in the missing data when they become available.
- Support for simulation at all levels of detail and at mixed levels of detail.
- Automation of tedious tasks.
- Monitoring of any object identified as an interface to confirm that the definitions are consistent.
- Ability to cut portions from existing designs and paste them into the current design.
- Automatic generation of documentation from the design definition.
- Indexed access to electronic copies of standards, requirements, guidelines, handbooks, preferred parts lists, and other documentation.
- Running totals for quantitative attributes of the design, such as failure rate, weight, power dissipation, and LCC.
- Prevention against loss of work if the system goes down.
- Exploration of a design alternative and return to original status if the alternative is rejected.
- Effortless mapping between logical and physical representations of the design.
- More capable layout and routing tools.
- Automatic computation of trace capacitance, including differences in capacitance as a function of layer.

Note. This table is taken directly from the RAMCAD Interim Report on Research and Development: Analyze Current CAD Capabilities (BCS, 1988).

Specifying the Reliability, Maintainability, and Supportability Design and Analysis Needs

To help identify where additional research was needed and the specific areas that should be investigated, shortcomings in design tools, methods, and techniques found in the RM&S design area were categorized. Although data was collected on the recognized problems of data-sharing, communication, and task-sequencing, an attempt was made to go beyond these problems by also collecting data in the following three areas: (1) understanding how the decisions of design engineers and RM&S specialists interact to determine the RM&S of the design, (2) identifying the activities and tasks that play the most significant roles in determining the RM&S of the product, and (3) understanding how the design engineer can be assisted in designing more reliable, maintainable, and supportable systems.

All information was placed in one of two areas: (1) RM&S design practices and (2) RM&S technology needs. A determination was then made as to where the research time should be spent based on an assessment of the available skills and the available resources. This information (summarized below) was documented in an interim report titled, "RM&S Areas to be Addressed" (BCS, 1989).

Reliability, Maintainability, and Supportability Design Practices

In the basic design process, detailed specifications of a product to be designed are derived through a top-down refinement of system concepts and requirements. As the design evolves, design engineers evaluate design alternatives for their impact on major design criteria. The best alternatives are used as the basis for the next iteration of the design cycle. The RM&S of electronic systems depends to a large extent on the physical characteristics defined during this process. For example, the reliability of an electronic assembly depends on the hardware implementation of its functions, including the quality of the components, the quality of the manufacturing process, its operating environment, and the margin between the operating demands placed upon individual components and their maximum ratings (usually called the derating factor). Similarly, the maintainability and supportability of a design depend on the characteristics of the hardware and the manufacturing process used to fabricate it. Thus, design engineers must consider implementation issues early in the design process in order to address RM&S concerns effectively. Unfortunately, design engineers do not always consider these items as important as many other design criteria (e.g., cost, schedule, and performance).

For DoD programs, government military standards (MIL-STDs) or system specifications define the basic RM&S tasks that design engineers must perform during the design process. Design engineers must conduct the RM&S tasks in accordance with MIL-STD-785B for reliability, MIL-STD-472 for maintainability, MIL-STD-1388-1A and MIL-STD-1388-2A for supportability, and MIL-STD-2165 for testability. Generally, design engineers accomplish each MIL-STD task to some degree during the design cycle, although their effectiveness and impact are usually less than optimum. Because of the lack of effectiveness, and a lack of focus on RM&S characteristics during the very early design tasks, design engineers cannot optimize a product based on RM&S criteria. Instead, design engineers modify the design late in the design cycle to ensure that the product will meet the minimum requirements.

RM&S allocation occurs throughout the system design process. System engineers derive the RM&S requirements at the subsystem and lower levels from the system-level requirements at the same time they derive the performance and cost requirements. They usually obtain the data to

support the allocation process from military sources such as the Navy 3M program and the Maintenance On-Line Data Acquisition System (MODAS) or commercial databases and component manufacturers. These sources provide ballpark RM&S estimates for current technologies based on similar systems or manufacturer testing. The experienced system engineer can use this data to determine realistic allocations for the detailed design requirements. However, the use of a new technology in a design forces the system designer to make assumptions about the capabilities of the technology. These assumptions can alter the overall result of the design. Ideally, as the design engineer designs the system and determines better estimates based on a more detailed definition of the design, reallocation of each design resource should occur as required to ensure the design reaches an optimal allocation mixture of the requirements. Doing so enables the design team to reallocate the design resources to optimize the design according to the greatest needs as based on updated and realistic design information. In practice, however, the system engineers cannot revise the allocations to reflect the evolving design state because of the cost and resources required to do so. They can only revise the allocations when the design cannot meet a requirement.

Reliability, Maintainability, and Supportability Technology Needs

The "RM&S technology needs" question was approached from two different, but related, perspectives. One approach was to investigate the RM&S design and analysis needs associated with assisting the engineer in optimizing the design for competing performance, cost, schedule, and RM&S requirements. The second approach was to investigate how to facilitate or automate the RM&S tasks specified in the MIL-STDs. Thus, the first approach looked at supporting the design engineer, while the second looked at supporting the activities of an RM&S engineer.

The specific RM&S analyses that need to be employed during the design process depend largely upon the requirements a system is expected to satisfy. For example, the design of a highly fault-tolerant, degradable system involves the use of modeling and analysis techniques (e.g., Markov models and/or Monte Carlo simulations) not needed in the design of systems with less-stringent design requirements. These considerations make it difficult to specify RM&S technology needs that are generally applicable.

The technology needs were separated into four distinct areas to facilitate the analysis: (1) general, (2) reliability, (3) maintainability, and (4) supportability. These needs were identified through interviews with the Boeing design community and analysis of the Computer-Aided Acquisition Logistics Support (CALS) Reliability and Maintainability (R&M) Summer Study on Complex Electronics (CALS Summer Study, March 1988). The needs briefly described below are more thoroughly discussed in a separate report (Kitzmiller et al., 1993).

Correcting the general needs should result in a design environment capable of facilitating a systematic exploration of the design space. Elements of a proper design environment include:

- support for a systematic design process in which integrated design tools support the evolution of the design from abstract concepts to a final, detailed definition of an item to be manufactured;
- a design database that captures and makes accessible all key design information required throughout the design process, and includes modifications (and the reasons for them) made after the system has been fielded;
- accurate predictions of all design metrics (e.g., RM&S, performance, and cost) in a timely manner early in the design phase;
- access to lessons learned through a database that is easily accessible to the engineering design community;
- capabilities to monitor the design for compliance with all applicable standards; and
- computerized data books that are in an easily understandable format, available to the design engineer at all times, and usable by design tools.

Reliability needs were divided into five key areas. These areas show where automated tools or improved methods could help the design engineer create a more reliable design. The areas where help is most needed include:

- performing reliability predictions and modeling earlier in the design cycle;
- forming a coherent and system-wide method of failure management and testability in which the impact of common failure modes on system operation are considered early in the design process;
- allocating and reallocating reliability requirements, objectives, and hardware resources throughout the design cycle to reflect the current state of the design and to ensure realistic allocations are derived and maintained so that an optimized design is achieved;
- monitoring a design for compliance with allocated reliability requirements, design practices, and guidelines throughout the design cycle and at all levels in the design hierarchy; and

- identifying modifications to a design, with a minimum of assistance from a reliability specialist, that cost-effectively enhance its reliability while not adversely impacting other design metrics.

Similarly, the maintainability needs were divided into six key areas. These areas deal mostly with human factors, preventive maintenance, and fault isolation and testing, with the last area having the most significant influence on current maintenance and support costs. As with reliability, the areas show where automated tools or improved methods could help the design engineer create a more maintainable design. The areas needing help include:

- performing maintainability predictions and modeling earlier in the design cycle;
- allocating and reallocating maintainability requirements throughout the design cycle to reflect the current state of the design and to ensure realistic allocations are derived and maintained so that an optimized design is achieved;
- monitoring a design for compliance with allocated maintainability requirements, design practices, and guidelines throughout the design cycle and at all levels in the design hierarchy;
- identifying modifications to a design, with a minimum of assistance from a maintainability specialist, that cost-effectively enhance its maintainability while not adversely impacting other design metrics;
- assisting the design engineer in developing effective test capabilities; and
- assisting the design engineer in creating designs that are more human-factors-oriented and, thus, easier to work on (e.g., improve access to better connectors to ensure there are fewer bent pins).

Supportability needs were divided into four key areas. These areas are similar to those defined for both the reliability and the maintainability needs. However, supportability issues often deal with such items as standardization, modularity, commonalty, and future availability of parts. These issues can affect both design reliability and design maintainability; however, R&M specialists do not manage these aspects of the design. The supportability needs include:

- performing supportability predictions and modeling throughout the design process;

- allocating and reallocating supportability requirements throughout the design cycle to reflect the current state of the design and to ensure realistic allocations are derived and maintained so that an optimized design is achieved;
- monitoring a design for compliance with allocated supportability requirements, design practices, and guidelines throughout the design cycle and at all levels in the design hierarchy; and
- determining the supportability implications of specific design alternatives or identifying modifications to a design, with a minimum of assistance from a supportability specialist, so that they cost-effectively enhance its supportability while not adversely impacting other design metrics.

Specific Research Area

After analyzing the above data, it was determined that the remainder of the research should focus on the area of preliminary unit⁵ design for reliability and testability. The objective was to develop and demonstrate an approach to providing the design engineer assistance in maximizing the reliability and testability of an avionics unit consistent with other applicable design goals and constraints. This approach was appropriate for the following reasons.

- It was consistent with the overall objective of investigating methods to aid in optimizing a design for performance, cost, schedule, reliability, maintainability, and supportability.
- Existing design tools do not properly support preliminary design. Most of the tools support the later phases of detailed design. However, it is possible to adapt the current tools to preliminary design which would create an evolutionary change in the design cycle.
- Decisions made during the preliminary design stage have a large impact on the performance, cost, and RM&S of a design.

⁵ As used here, unit refers to an avionics module consisting of several circuit boards. Unit is used rather than line replaceable unit (LRU) because the current discussion assumes a two-tier maintenance approach. In addition, preliminary unit design refers to those tasks performed at the very end of system engineering and at the beginning of detailed design. It includes functional analysis of the unit, design creation and evaluation, and preliminary partitioning to the circuit-board level.

- Preliminary design requires expertise in both systems engineering and detailed design. Very often there is a problem with "handing off" the design between these two areas. Since few design engineers are knowledgeable in both areas and in working designs as they cross this imaginary boundary, effective design tools would be particularly valuable.
- Testability in one form or another is the major factor determining the maintainability of electronic systems.
- This approach complemented the work of other RAMCAD contractors.

It was also decided that the tools created during the second phase of this research effort should take advantage of advanced artificial intelligence (AI) and expert system techniques. Use of AI and expert system techniques would allow the tools to be sufficiently sophisticated to provide on-line expertise to the design engineer. Further, it was decided that the design environment in which these tools would be implemented should provide a consistent interface for all the tools and a single internal design representation. The system should also preserve not only the formal design documents, but the choices that were considered and the rationale for rejecting some decisions and accepting others. Lastly, for the tools to be truly effective, they must contain two types of design knowledge: knowledge about designing and knowledge about the design.

Ideally, the tools should express knowledge about designing through some form of design advice. They should offer information about the design process (e.g., what step is usually performed next and what tool would normally be most helpful), monitor analyses in progress to make suggestions where relevant, and suggest or criticize approaches taken by the design engineer based on attributes that determine good or bad designs. These functions are not available in any commercial tools. Currently, it is entirely the design engineer's responsibility to provide knowledge about designing.

Knowledge about the design should be expressed through stored data about the design in process and a historical account that is updated at major forks in the design path. This data includes the alternatives, choices made, any factors influencing the choices and their relative weights, and the reasons the design engineer accepted or rejected each alternative. The design engineer must provide at least a portion of this data; however, the CAD/CAE system could obtain some of this data from the output files of analysis tools after the design engineer runs an analysis and implements the results.

Eventually, after a group of design engineers create numerous different designs using the CAD/CAE system and associated tools, they would have access to previous similar designs.

These designs would include those that showed promise but could not be used because they did not fit system requirements. The tools could provide intelligent access to these designs, supporting reuse of previous designs and early estimates of properties such as heat dissipation and cost as well as field data of the system if the design progressed that far. The use of this field data, compared to design estimates, would help the design engineer determine where to alter an existing design to meet current design specifications.

Developing a Casebook

To gain knowledge about designing, the next part of the research effort focused on determining what increased or decreased the RM&S aspects of a design. A list was compiled of design RH&Gs that the RAMCAD working group recognized as either enhancing or degrading the RM&S of an electronic system. The attempt to go beyond the data normally found in literature required repeated interviews with the Boeing specialists selected for this part of the research.

The data gathered was compiled into a draft "casebook." This casebook was reviewed by a select subgroup and, subsequently, delivered to the government as an interim report. The RH&Gs listed in the casebook were divided into several sections. The sections included general topics as well as circuit design, part selection, and built-in test. In addition, each RH&G mentioned in the casebook states the design functions, tasks, and design characteristics it affects (e.g., reliability, testability, repair quality, and performance). This data was written in a form that can be electronically integrated into a CAD/CAE tool.

Great care must be taken when using data like that in the casebook. Each RH&G has a specific purpose. As the circumstances surrounding the design process change, the design engineers must review the RH&Gs to ensure they are still valid. Some rules in the casebook are in effect basically because of shortcomings in the current CAD/CAE tools or the design process. Thus, the RH&Gs may not point towards an ideal solution but to one that is reasonable based on those CAD/CAE features available when they were written. Also, RH&Gs should only be used to help guide the design engineer. Using them to generate a set of possible solutions removes part of the design engineer's ability to use common sense to ensure the RH&Gs are applicable for the design problem being solved.

Developing Adjudication and Sequencing Rules

In conjunction with creating the casebook, there was a requirement to create adjudication and sequencing rules that would implement each part of the casebook through the software tools.

These rules were created to help identify when the software tools would use the different RH&Gs and to provide guidance when arbitrating those situations where more than one rule, heuristic, or guideline is applicable.

Creating proper sequencing rules that would invisibly (to the user) direct how the software tools used the RH&Gs throughout the design cycle required knowledge of a good representation of the design cycle. Naturally, the one created under this research effort was used to ensure a smooth integration of the methodology and tools. The sequencing rules can be viewed as the process of taking data from an electronic casebook, determining which data apply during the current phase of the design cycle, and applying those RH&Gs in an organized way.

The adjudication rules, on the other hand, determine which RH&Gs are most important when more than one is applicable. Developing this information is difficult because many of the RH&Gs conflict. One example of this occurs when considering the modularity, LCC, testability, and reliability factors affecting a design. Partitioning to fewer circuit cards makes isolation easier, increases the probability of repairing a module on the first try (increasing testability metrics), and reduces the chance of faults caused by circuit card connections (increasing reliability metrics). However, it could increase spares costs and would reduce the likelihood of using disposable modules (possibly increasing the LCC). Notably, partitioning to more circuit cards would have the opposite effects on all the metrics.⁶ Thus, using the proper casebook data based on the relative importance of each of these factors is very difficult. It requires that the tools be intelligent and utilize information that can only be determined through a proper system engineering analysis.

Developing the Design Methodology

Next, how the information gained in the above tasks could be used in a design methodology to improve design RM&S was investigated. The methodology that was created is briefly described below and more fully defined in another report published by AL/HRG (Kitzmiller et al., 1993).

The current detailed design process is shown in Figure 5. However, this figure does not show the problems associated with current design practices. The research demonstrated that current designs are often excessively constrained in an attempt to avoid "undue" risks. Risk aversion techniques force system engineers to create designs that meet the requirements based on current

⁶ This example lists only a few items that would actually impact the decision if a complete set of rules were incorporated. Other factors that could be involved include fewer spares requirements because of a more accurate diagnosis of a single card and the ability to test individual components on a larger card if repair is to be performed at that level.

technology more often than trying to optimize a design based on technology expected in the near future. Also, by attempting to avoid risks, system engineers usually create very tight requirements and specifications that limit the detailed designer's ability to optimize the design. When these problems are combined with the rapid advancement in current electronic, manufacturing, and computing technologies, a situation results in which it is difficult for detailed designers to optimize a design because they cannot take advantage of future technology or they are too confined by the design requirements. Thus, the product will meet all the customer's specifications instead of being the best possible product to meet the customer's needs.

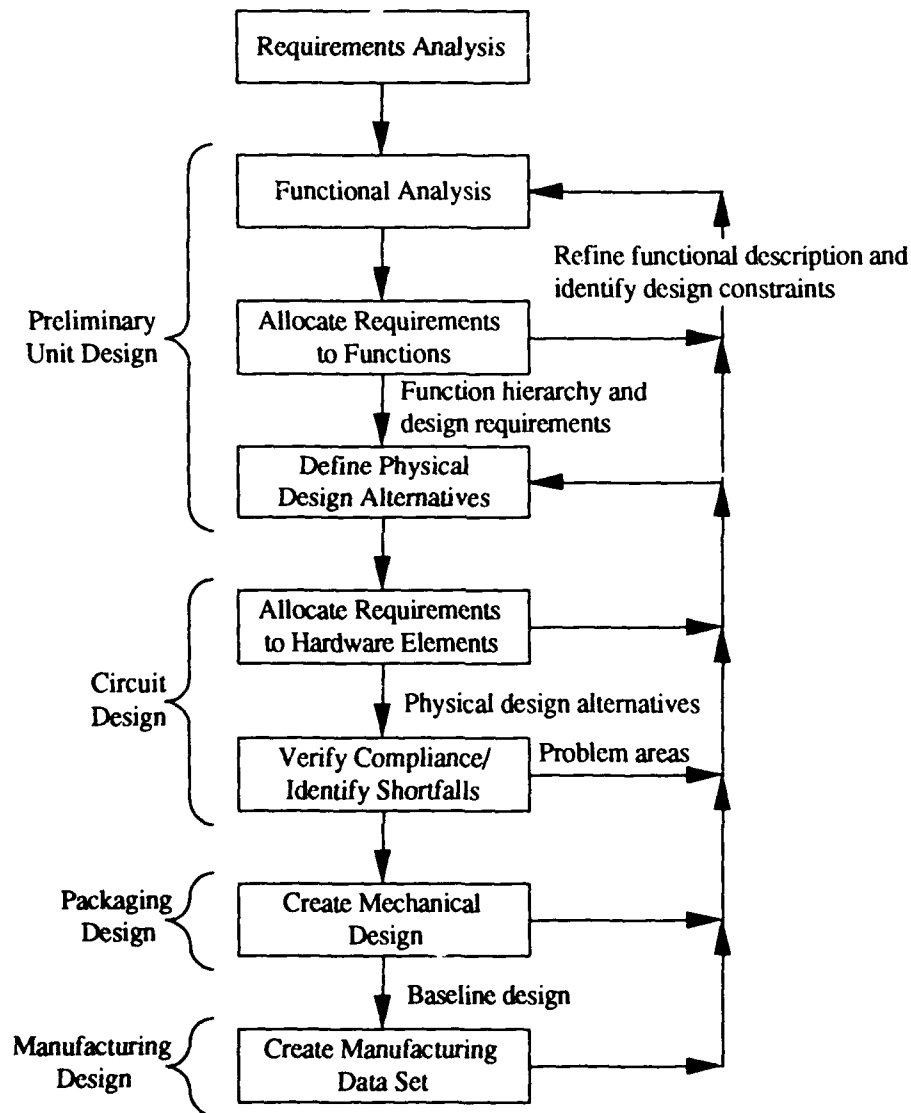


Figure 5
Current Detailed Design Process

This problem leads to the obvious answer of keeping options available as long as possible throughout the design process. Doing so provides two benefits. First, the design engineer can insert new technology when it becomes feasible instead of being forced to use older technology. Second, the length of time between decision-making and fielding the system will be shortened because the decisions will be made later in the design cycle. Based on this information, the decision was made to modify the design process as detailed below, with the understanding that the main thrust during each design task is to create optional designs and select the best alternatives without eliminating future options. For ease of explanation, the methodology is only discussed in the context of the detailed design process. However, this methodology was designed for use during both the system engineering and detailed design phases. This was a primary goal of the RAMCAD working group because it is important to increase RM&S considerations in the early phases of design and to continue supporting their consideration throughout the design cycle. In addition, it was decided that decisions should not be made as early as they are made currently. Therefore, a basic change is necessary in the tool capabilities and other resources presently available to design engineers. The design tools must provide the design engineers with the capability to analyze their options without access to all the data currently required during design analyses. Also, the design engineers need more resources throughout the design cycle to ensure they can analyze their options at each level in the design hierarchy instead of latching onto the first option that meets the requirements.

The stages of the design process affected by the methodology are the preliminary unit design and circuit design stages because, as stated earlier, there is currently very little help at these stages. Thus, this research effort could have the greatest impact in these stages. Three important considerations were used to develop this methodology. The first was that design engineers should not begin the initial design and trade-off analyses assuming every factor should be taken into account. Instead, first attempts should use only a few important design aspects based on a weighting of the relative importance of all design factors. In other words, assume that 20 percent of the factors account for 80 percent of the design requirements and decisions. If the design engineer uses just that 20 percent, the initial analyses will be quicker and the search for alternative designs can include more possibilities. This would help design engineers focus on those parts of the design that should yield the most benefit and encourage them to consider more design options which would yield better decisions through the trade-off analyses.

Second, a purely numeric process could not be relied on during the design analyses because, currently, such processes do not properly account for qualitative factors. In addition, not enough is known about the failure mechanisms of some hardware parts (e.g., connectors) to ensure that a

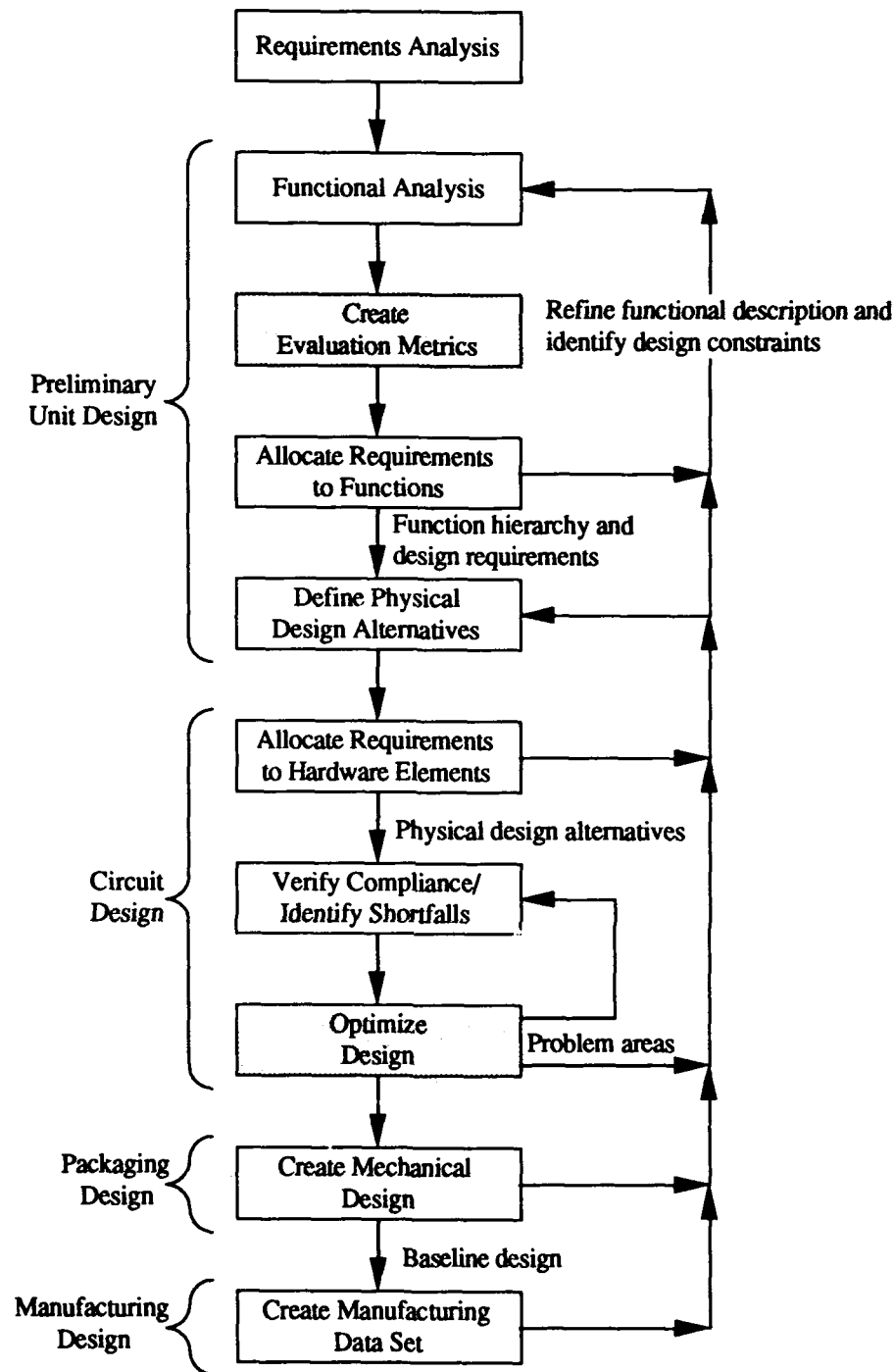
numeric approach takes them into account properly. Thus, the design analyses must include the design engineer's knowledge.

Third, although tasks might be added to the design process, any attempted improvements must ensure that the design will improve sufficiently to lower the number of iterations required to fix problems later in the design cycle. This would ensure that the overall time required to create the design would not increase significantly and might decrease. However, the increase in product quality, decrease in LCC, and subsequent increase in customer satisfaction would be worth any small increases in the design cycle time.

The methodology does require the design engineers to perform tasks that they may not now perform (Figure 6). The additional tasks, when performed with the aid of the appropriate CAD/CAE tools, would allow design engineers to formulate a strategy for designing the electronics that optimizes a design based on its most important aspects.

The first additional task, explicitly defining the relative importance of the design criteria and creating an evaluation metric, requires that the relative importance of design requirements be established, and that the design parameters and evaluation metrics the tools will use as the basis for evaluation be defined. Once the design engineer ranks the requirements, the relative importance of each is determined and evaluation metrics are created for use during design optimization. The metrics provide a mathematical weighting of various specification measurements (e.g., mean time between failure [MTBF], throughput, and mean time to repair [MTTR]) to show the relative importance of each design requirement. The design engineer will use the metrics later in the design cycle to help differentiate between design options. However, if the metrics do not adequately differentiate between the alternatives, this step can be reaccomplished to ensure the validity of the criteria and to add additional criteria, as appropriate. These metrics, which can be both quantitative and qualitative, are then inserted into the CAD/CAE tools for later use in design optimization.

The second task, optimizing the design, can use the intelligent CAD/CAE tools designed under this effort, as well as others, to help fully explore the design space available to different design options. It is assumed that the design engineer has created more than one viable design alternative and now wants to determine which is best based on customer needs. However, these tools would also be useful if the design engineer has created only one design option. The tools are meant to use the knowledge gained in the earlier research into design knowledge to help the design engineer determine which design techniques to apply to the different parts of the design under analysis and what the results would be if one or more techniques were applied. To understand how this is



Note. Although the Verify Compliance/Identify Shortfalls Task is shown before the Optimize Design Task, these two should be performed almost in parallel. The first time the design reaches the Verify Compliance/Identify Shortfalls Task, it must be checked to ensure it meets minimum standards. Then, each time a change is made to help optimize the design, the design must be reverified. This ensures that the design always meets its requirements (e.g., cost, performance, and board area).

Figure 6
Proposed Detailed Design Process

possible, imagine the design solution space as a region in n -dimensional space with numerous discrete subsets of that space containing groupings of similar design solutions. If the design engineer only finds one of these solution subsets, the tools could be used to explore the boundaries of the space defined by that subset to determine the optimum design solution for that subset.

Although both of the above tasks are shown as performed during a specific time in the design process, it is expected that design engineers would actually perform them whenever appropriate. For instance, creating and updating the evaluation metric should ideally start during requirements analysis. However, it is shown later in the model because the first cut at the metric should be finished by this point in the design cycle. Furthermore, both tasks are shown as single steps to ensure they are accomplished at least once during each design phase. As stated earlier, the descriptions are written as if these tasks would be performed only during the detailed design phase. However, note that the tasks added to the detailed design phase are often considered system engineering tasks. That is, the design engineer is asked to create design options and explore the boundaries that restrict each option so that the best design solution is used to create the product. Likewise, the system engineer is being asked to perform extra work by determining the importance of the design requirements and exploring the boundaries of more design options. Lastly, this process will likely force the detailed designer and system designer to improve their interaction. Thus, the detailed designer will gain a better understanding of the importance of various criteria and of the effects detailed design decisions have on system design.

III. SOFTWARE TOOL CODING

The second phase of the research effort focused on creating proof-of-concept CAD/CAE software tools capable of performing those tasks required to demonstrate the feasibility of the methodology. Four tools were created to perform this work: (1) System for the Interactive Design and Computer Analysis of Reliability (SIDE CAR), (2) Inherent Testability Analyzer (ITA), (3) Statistical Testability Analyzer (STA), and (4) Scan Assistant (ScanIt). The concepts behind and use of each of these tools is discussed below. The Phase 2 schedule for the development of these tools is shown in Figure 7.

To create these tools, a design environment was needed that provided access to ECAD/ECAE tools and in which the proof-of-concept tools could be quickly coded. The Symbolics Lisp environment with the NS ECAD system was chosen to perform this work. The NS ECAD system provides schematic and design capture, design representation, and other standard ECAD tool capabilities. The Symbolics Lisp environment is not as widely used as other design environments;

however, the designers of NS ECAD provided access to the system source code and agreed to assist in modifying a version of the system to incorporate the proof-of-concept tools. The proof-of-concept tools were intended to facilitate the research; therefore, it was determined that this level of support was crucial if a working research environment was to be developed in a relatively short period with limited resources. This support allowed for the implementation of many capabilities that could not have been added otherwise. One such capability allows the SIDECAR and ITA tools to implement specific design techniques (e.g., triple-modular redundancy [TMR] and the addition of error-detecting code) and to "know" that certain design changes are required (e.g., additional board area is consumed and the reliability statistics are affected). In addition, the NS ECAD system "knows" to modify the schematic and design representation to include the changes required by the implementation of the design technique.

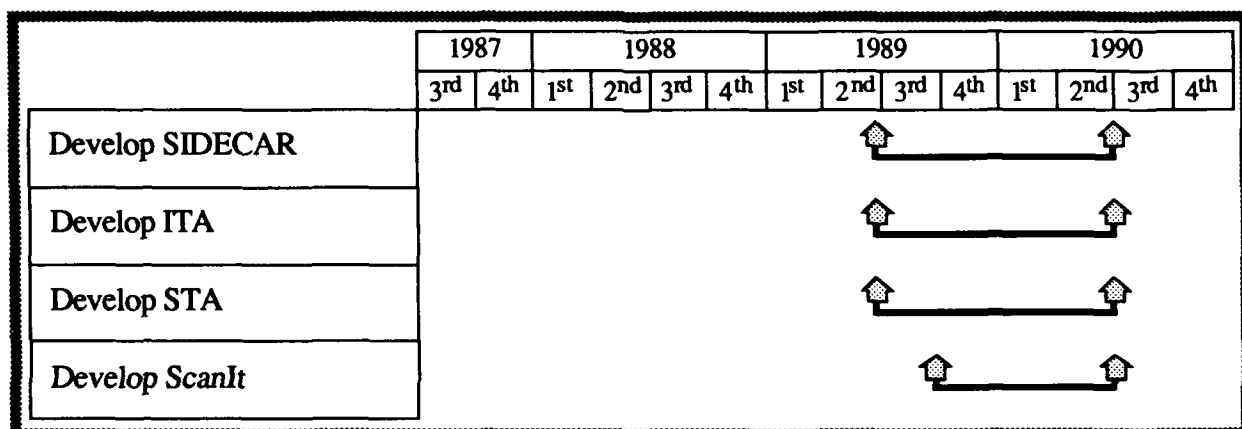


Figure 7
Phase 2 Schedule

Developing the System for the Interactive Design and Computer Analysis of Reliability

Given specific fixed functional requirements, the design engineer can rarely improve the reliability of a design beyond that provided by the inherent quality of its components without using one or more of the techniques listed in Table 2. However, improving design reliability to a required level is not as easy as just applying some of the techniques to the design. Burdens (e.g., weight, cooling power, and cost) are associated with each technique and the design engineer must determine which combination of techniques will provide the most benefit while minimizing the additional burdens. SIDECAR was developed to aid the design engineer in assessing the relative benefits and burdens associated with alternative reliability enhancement techniques. SIDECAR provides aid through analysis results which the design engineer can review in a variety of formats.

The different formats were created based on the design engineer's need to determine which, where, and to what extent each design technique should be applied.

Table 2. Reliability Enhancement Techniques

Category	Technique
Fault Avoidance	<ul style="list-style-type: none">• Reduced electrical, mechanical, and thermal stress levels• Upgraded component packaging and quality• Increased component integration level• Environmental control features
Fault Detection	<ul style="list-style-type: none">• Duplication and compare• Error detection codes• Consistency and capability checking• Time domain detection
Static (Masking) Redundancy	<ul style="list-style-type: none">• N-modular redundancy with voting• Error correction codes (hardware or software)• Interwoven logic• Coded-state machines
Dynamic Redundancy	<ul style="list-style-type: none">• Reconfiguration• Standby sparing• Graceful degradation

Note. This table is taken directly from the RAMCAD Interim Report on Research and Development: RM&S Design Techniques (BCS, 1990b).

Based upon the analysis of needs (BCS, 1989), it was determined that a decision support tool to assist in designing for reliability needed to provide the following capabilities:

- reliability modeling,
- automated sensitivity analysis,
- support for user-defined correlations and comparisons,
- methods to identify and rank the major contributors to system unreliability, and
- methods to identify and rank design alternatives and the design changes that can potentially improve the reliability of a design.

SIDECAR was designed to support a propose-evaluate-revise approach to reliability. First, the design engineer must propose a baseline hardware implementation and alternative implementations of the unit functions. Next, the design engineer, with the help of SIDECAR, evaluates these

choices relative to one another and to the requirements. Finally, the design engineer modifies the baseline and, if required, proposes additional design alternatives based upon this evaluation. The purpose of SIDECAR is to provide reliability estimates to aid in the evaluation of designs and design alternatives, assist in the identification of major contributors to design unreliability, and aid the design engineer in identifying modifications to a design or design element that improves its reliability.

SIDECAR was created by Charles Yount and Dr. Daniel Siewiorek at CMU as part of this research effort. It was coded in C++ and, for this effort, resided on a Sun workstation networked to a Symbolics workstation. SIDECAR is a stand-alone program that the design engineer can access through either its own user interface or one developed for a CAD/CAE system. The user interface employed for this research effort was coded in Lisp and resided on a Symbolics workstation as part of the NS ECAD system. SIDECAR was designed to take advantage of the capabilities of current reliability analysis tools that take part reliability figures and roll them up to determine unit, subsystem, and system reliability metrics. Thus, SIDECAR is an addition to the suite of tools available to the design engineer; it is not meant to replace the current reliability analysis tool. For this research effort, a reliability prediction tool called Lambda, also created at CMU, and an Informix database were used. The database contains the electronic parts information required for SIDECAR to perform its analyses. SIDECAR does not work autonomously; rather, it provides reliability information to the design engineer and recommends design changes that can be accepted or rejected. SIDECAR bases its recommendations on the results of the set of exploration routines described below. For SIDECAR to run the following routines, the results from a reliability evaluation must be available in an electronic format. In addition, because SIDECAR requires updated analysis results during some of its routines, it must be able to automatically invoke the reliability analysis tool before or during any of the routines.

Exploration of Parts

The exploration of parts routine seeks to identify design component changes that would improve the reliability of the design. It searches the electronic parts database for higher quality parts and components which the design engineer may substitute for parts or components in a design. After performing a search, the routine provides a ranked list of possible part and component changes. The routine bases the ranking on the projected impact each change would have on a metric or set of metrics and other information about the criticality and "role" of the components. The metric or set of metrics takes into account a set of design criteria (e.g., reliability, cost, and board area) identified by the design engineer at the beginning of the routine (i.e., presumably defined during the create evaluation step shown in Figure 6). The design

engineer should base the metric or set of metrics upon the relative importance of the design criteria. The version of the NS ECAD system used during this effort can include the criticality and "role" of the components (e.g., the component is part of the core functionality of the subsystem versus an extra part required for the testability of the subsystem) in the design representation. SIDECAR is not restricted to looking only for direct part-to-part changes. The program may look for something as similar as a ceramic chip to replace a plastic chip, assuming the functions are exactly the same. However, if an application-specific integrated circuit or even a circuit board is identified in the parts database as a functional substitute for a collection of discrete gates, SIDECAR will include it on the list of possible substitutions.

Exploration of Temperature Changes

The exploration of temperature changes routine helps the design engineer determine where active cooling techniques may be beneficial by performing a temperature sensitivity analysis on all or part of a design, as specified by the design engineer. The routine performs the analysis by determining a failure rate prediction for each component at a variety of temperatures and then computing the change in the metric resulting from a failure rate reduction. The design engineer can define the temperature range for this analysis (maximum, minimum, and delta temperature) or SIDECAR will default to predetermined reasonable values. The routine displays the analysis results as a list of design elements ranked according to their temperature sensitivity. This allows the design engineer to make a quick determination of the potential effects of cooling and other changes in the operating environment.

Exploration of Marginal Benefits

Table 2 includes some other techniques used to increase system reliability besides parts replacement and temperature changes. However, it is not always intuitively obvious whether and where to use each technique in a design, especially for the apprentice design engineer. Even a table that ranks the parts and subsystems of a design by their contribution to the hazard rate does not provide all the data needed. SIDECAR provides one possible solution through an exploration of marginal benefits routine to aid the design engineer in determining which components and parts of a design will yield the greatest improvement in system reliability, assuming an improvement in reliability is physically feasible and does not exceed design criteria. This routine determines the maximum improvement in the reliability of any level in the design hierarchy, assuming the reliability of its components can be improved to a user-specified level. It explores the design hierarchy in a top-down manner, further evaluating subcomponents of a component that offer the greatest possibilities for reliability improvement. This allows the design engineer to determine the

marginal effect of each part and subsystem in the design hierarchy. Thus, any changes to the design should have the maximum impact on the entire subsystem under analysis.

This routine was designed to be general purpose so that the design engineer can perform this analysis for any design characteristic (e.g., testability and maintainability). Once the design engineer knows where improving a particular design characteristic will provide the most benefit, other techniques, such as those in Table 2, can be used.

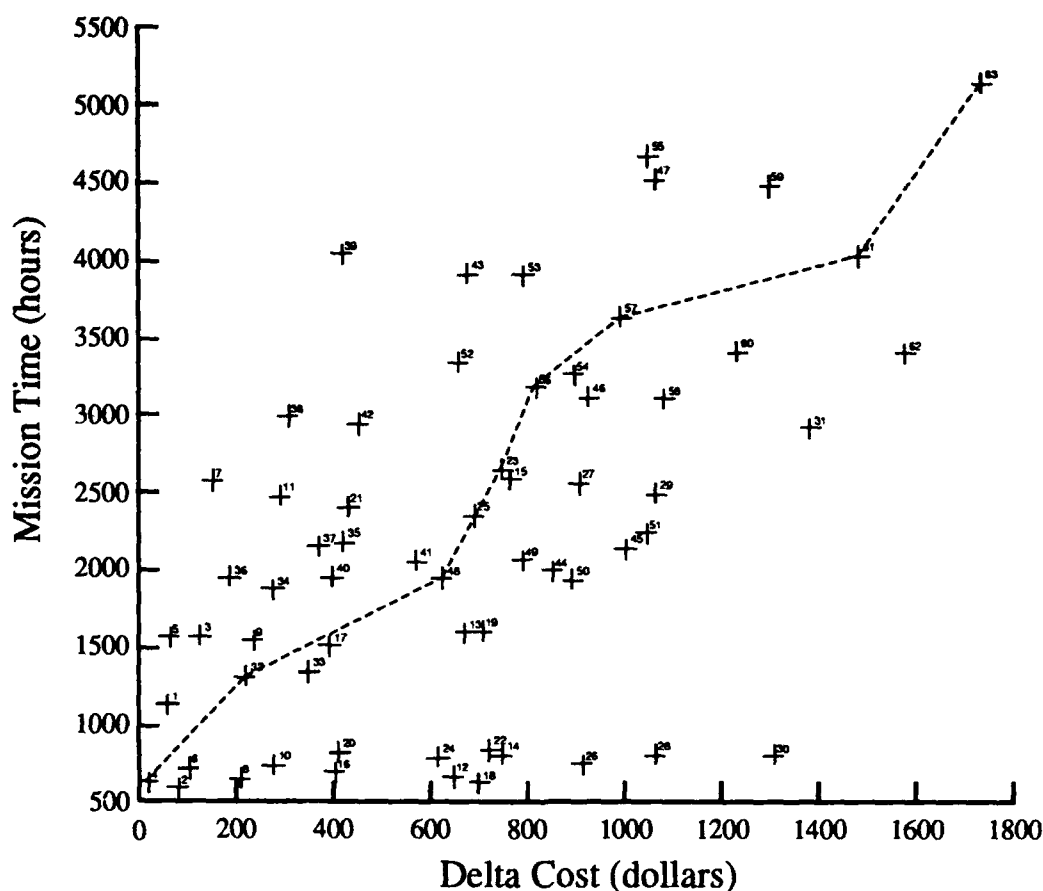
Exploration of Combinational Changes

The exploration of combinational changes routine is the final general purpose routine that allows the design engineer to select those part changes, temperature changes, and other modifications that SIDECAR determined to be beneficial and to determine the combined effect of any subset of these changes on various design metrics (e.g., mission time, MTBF, cost, and board area). For example, if six possible changes are being considered, the design engineer can define the alternatives through the SIDECAR user interface before changing the design. The tool will analyze the 63 possible combinations of alternatives and provide a graphical or tabular display of the impact each combination would have on the design metrics. One such graphical output is shown in Figure 8. From this graph, the design engineer can quickly determine which combination of changes will be most beneficial based on the design constraints. This allows the design engineer to implement *only those changes that conform to the design constraints and provide the greatest payoff.*

Developing the Inherent Testability Analyzer

Currently, there is a lack of CAD/CAE tools to effectively support testability analysis before the availability of completed detailed circuit diagrams. This is caused by a mistaken belief that no testability analyses can be performed until the design is thoroughly defined, including test vectors and other requirements of current testability analysis tools. In addition, testability analysis is considered infeasible early in the design cycle for three main reasons: (1) insufficient time to develop detailed tests early in the design process, (2) slow and expensive simulation technology, and (3) unjustifiable costs in both the money and the time required because of probable changes in the design or its requirements later in the design cycle. However, the assumptions behind these reasons do not take into account that design engineers and testability experts could apply some testability techniques to partial designs to determine a rough estimate of their overall testability. The design engineer can then use these rough estimates to determine the best design approach to solve the problem. Instead, the lack of early testability analyses causes the design engineer to

release the design for hardware prototype with limited, inadequate test analyses. Consequently, hardware prototypes usually have poor testability which, in turn, causes two problems: design errors and prototype malfunctions are more difficult to debug, and testability requirements of the design cannot be verified.



Note. This figure was copied from a paper written by Yount & Siewiorek (1991). The dotted line represents the path that might be taken by a computerized reliability improvement program that did not first analyze all combinations of alternatives but just took each alternative into account based on the expected individual increase in the reliability metric.

Figure 8
System for the Interactive Design and Computer Analysis of
Reliability Sample Output for the Exploration of Combinational Changes

ITA was created to help rectify these problems. ITA was coded in Lisp to run on a Symbolics workstation as an extra CAD/CAE analysis tool. Its goal is to provide a quick, low-cost estimate of the inherent testability of a design, or part of a design, during the first iterations of the design cycle based on the controllability and observability of the inputs and outputs of the design.

Inherent testability analysis is not new. Knowledgeable design and test engineers usually perform such an analysis before drawing release. Because it is heuristic and uses simplified device models, inherent testability analysis requires much less information than existing testability analysis tools and provides a rapid and relatively inexpensive way to evaluate the testability of an in-process design. However, this research effort did not turn up any commercial inherent testability analysis tools.

ITA was intended to fill this analysis gap by performing a relatively coarse analysis of the testability of a design. The design engineer can use it early in circuit design because it does not require any test vectors and it can be used on parts of a design instead of requiring that a whole circuit design be completed. Unfortunately, the current version has several limitations. First, it can only be used on digital designs. Second, it is not 100 percent accurate because it ignores consistency problems such as conflicts between signals necessary to observe outputs and signals necessary to control inputs. Third, component behavior models must be built for each component. This is simple for items such as AND and OR gates, but defining models for complex, off-the-shelf chips is much more difficult. Last, it predicts fault detection but ignores fault isolation.

ITA employs a testability information propagation algebra for electronic devices (P-algebra) developed under this research effort to perform its analysis. ITA uses P-algebra and component-specific heuristics to propagate and merge controllability, observability, and testability tokens throughout a design and determine the results. The design engineer can use these results to determine the overall testability prospects of the design and locate the positions of specific shortcomings in the testability of the design.

Like SIDECAR, ITA has a set of routines which the design engineer employs to create the analysis results. The following is a brief description of these routines.

Define Controllability and Observability

The first step requires the design engineer to define the controllability values of the inputs to and observability values of the outputs from the part of the circuit under analysis. The design engineer performs this step by assigning the appropriate P-algebra value to each circuit input and output through the ITA user interface.

Controllability Propagation

ITA performs the next step by propagating the controllability. It accomplishes this by inferring the controllability of all the pins based on the user-defined controllability, circuit connectivity, and

component controllability behavior models. The propagation is an iterative process in which ITA propagates the controllability through components and across wires to connected components. Iteration continues until controllability (which is monotonic) stops increasing.

Observability Propagation

The next step, observability propagation, must follow the controllability propagation step because the observability of any pin is based on the function of the component, the controllability of its inputs, and the ability to observe its outputs. Thus, if the controllability of the component is unknown, ITA cannot determine its observability. The observability propagation is algorithmically identical to controllability propagation, except that ITA uses observability behavior models and values instead of controllability behavior models and values.

Component Testability Analysis

After ITA performs both the controllability and observability propagation routines, it performs a testability analysis routine. This routine computes a testability value for each pin of each component being analyzed. ITA bases this value upon the testability model and the observability and controllability data for each component. Finally, ITA computes the average testability value of all the pins for each component and presents this data as testability measures of merit.

After ITA presents these results to the design engineer, the design engineer can easily determine why a component is not testable based on its controllability and observability values. The design engineer can then add built-in test equipment (BITE) as necessary to ensure the design reaches the proper level of testability.

Developing the Statistical Testability Analyzer

STA was created to assist in the optimization and allocation of testability during both the systems engineering and detailed design phases of design. It combines testability analyses, requirements, and allocations with failure rates and severity (as defined in MIL-STD-1629A) and uses this data to find and sort testability problems by their impact on cost, availability, maintainability, and mission success. STA is intended to help the design engineer define, allocate, and evaluate the cost and effectiveness of proposed test resources.

In the systems engineering phase, STA assists the design engineer in defining test coverage allocation data at abstract levels in the design hierarchy and in computing and evaluating fault detection and isolation statistics. During the detailed design phase, STA reports any differences

between predicted and allocated coverage. It supports these tasks by providing methods to estimate a variety of basic testability-related metrics and by assisting in the allocation and tracking of testability levels, generating fault catalogs, and estimating the effectiveness and cost of proposed fault detection and isolation features.

The STA process requires five steps: (1) input hardware test characteristics, (2) input test coverage data, (3) run STA analysis, (4) generate STA reports, and (5) make necessary design and test modifications. The process is iterative with the repetition of the third, fourth, and fifth steps until the design reaches the required testability level. The process is explained in more detail below.

Input Hardware Test Characteristics

STA requires specific hardware characteristic data to perform its analyses. The first part of the data is the component usage ratio. This ratio defines the portion of a component's failure rate that is relevant to test detection (i.e., the percentage of failures that will either adversely affect system functionality or cause unscheduled maintenance). This is important because a component may have either unused sections (about which SIDECAR can inform STA) or unrequired functions (about which the design engineer must inform STA). The second part of the data is an adverse BITE ratio. This ratio is the conditional probability that a section of a component used only for built-in test has failed in a manner that will adversely affect the operational system, given that the component has a fault. The third part of the data is the component failure rate computed by Lambda or a similar reliability analysis tool.

Input Test Coverage Data

The inputs to this task are the schematics of the hardware (taken from NS ECAD files) and a description of the tests to be used. These inputs allow the tool to determine which faults each test will detect (e.g., test "A" will detect faults 1, 2, and 3 while test "B" will detect faults 2, 3, 4, and 5) and, if detected, which tests will find each fault (e.g., fault 1 will be detected by test "A", faults 2 and 3 will be detected by tests "A" and "B", and faults 4 and 5 will be detected by test "B"). The design engineer can define the test coverage data at any level in the hardware hierarchy. The data can also be indexed for allocation or prediction and associated with any test mode.

Run Statistical Testability Analyzer

Through the STA analysis, 11 different statistics for individual components, groups of components, and tests can be determined. These statistics give the design engineer an assessment

of the test coverage and isolation of the design. Detected and undetected failure rates and the probability of fault detection (Pfd) are the primary statistics used to indicate the test coverage. Similarly, discrete isolation parameters, mean prioritized replacement position, and mean replacement list size are the primary indicators of test isolation.

Generate Statistical Testability Analyzer Reports

STA can generate sets of reports to make fault detection and isolation statistics available to the design engineer. In addition, these sets of reports can show the statistics for individual components, groups of components, or groups of tests. These reports can also compare the allocated requirements to predicted fault detection and isolation statistics.

Make Necessary Design and Test Modifications

Currently, testability experts perform the same type of test analyses as STA only after a design is complete and it has left the detailed designer's control. By then, breadboards are probably being created for testing purposes and external test hardware may be under development. If the analyses reveal serious problems, the testability engineers may need to stop these efforts and re-initiate the detailed design process. With STA, the detailed designer can accomplish these analyses before releasing the design. Thus, shortcomings can be determined and modifications made before any hardware prototypes are built. This method lowers the risk of impacting subsequent processes and creates the opportunity to enhance the fault coverage and isolation features of the design.

Currently, STA is not capable of helping the design engineer by recommending specific hardware or test changes to improve design testability. However, it can provide information to the design engineer about the effectiveness of the proposed fault detection and isolation features, and highlight areas in which the design is not tested adequately. STA can also provide a large percentage of the testability-related estimates needed by the design engineer to select the appropriate techniques to rectify testability problems.

Developing Scan Assistant

The large increase of circuit density in current designs has increased the intricacy and costliness of testing problems. To alleviate this problem, numerous design for testability (DFT) techniques have been created. While ITA and STA help the design engineer determine whether a design will exceed or fail to meet design criteria, neither helps in overcoming the shortfalls. Consequently, ScanIt was created.

ScanIt helps the design engineer employ the DFT technique called boundary scan. This technique improves testability by making the pins of functional components both observable and controllable; thus, it is the logical tool to combine with ITA. Boundary scan was used in this research effort because the general approach to the boundary scan technique can be useful at both the system engineering and detailed design levels of the design process.

The choice of technology to be used in a design, requirements to perform on-line testing of critical functions, repair strategies required, and the amount of modularity to be used can all affect the complexity of test development. However, the design engineer can use boundary scan to reduce this complexity. The concept of boundary scan, at its simplest and most expensive, requires that a scan register (i.e., shift register, latch, or boundary scan cell) be placed next to each functional component. Thus, signals between chips or at component boundaries can be controlled and observed using the scan design method. This can be quite expensive in monetary costs, board area, and other limited resources. Thus, design engineers usually design only a partial scan capability with a limited number of points designated for scan registers. ScanIt can help by recommending where these scan registers would be the most beneficial. These scan registers can then be used through either passive or active test techniques to determine the status of the system.

To perform its analysis, ScanIt needs a block diagram of the circuit specifying the functionality of component pins and the interconnection between components. ScanIt can then automatically place scan registers using the following practices:

- add scan registers to break feedback and thus reduce the complexity of test development,
- minimize the number of scan regions to reduce the cost of using boundary scan, and
- sequence scan regions to simplify fault isolation.

In addition, ScanIt allows the manual addition and deletion of scan registers. This, in turn, permits the design engineer to use other practices in determining where to incorporate boundary scan.

IV. EVALUATION AND VALIDATION

The third and final phase of this research effort required creating metrics and methods to evaluate CAD/CAE design tools and validate design methodologies in general and the tools and methodology created during this effort in particular. This phase was divided into three tasks that

included (1) developing generic evaluation criteria, (2) determining which criteria and methods should be used to measure the usefulness of the proof-of-concept tools and methodology, and (3) performing an evaluation and analyzing the resulting data. Figure 9 shows the Phase 3 schedule. In this section, the generic evaluation criteria created during the first task are described, the evaluation challenges and approaches determined in the second task are presented, and the criteria and methods defined in the second task along with the results of the third task are presented.

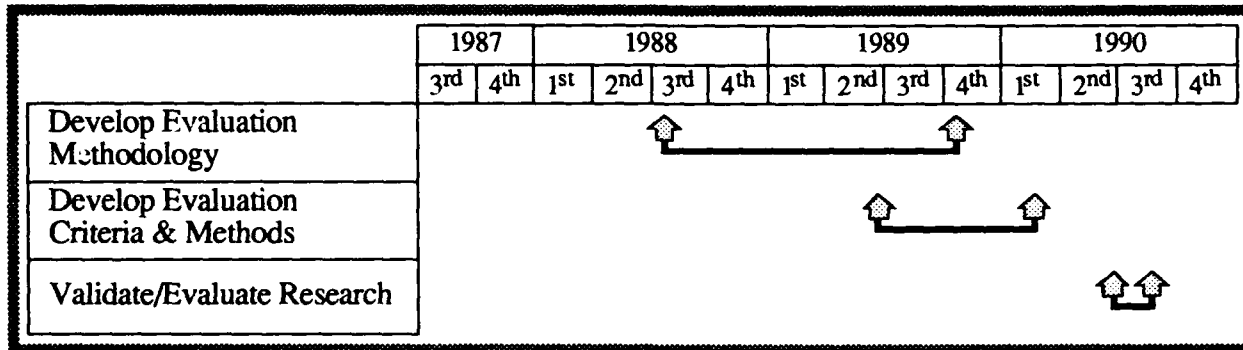


Figure 9
Phase 3 Schedule

Developing Generic Evaluation Criteria

To develop criteria that measured either new design methodologies or new CAD/CAE tools, information was gathered from two sources: (1) technical reports that discussed RAMCAD-like tools and analyses of any CAD/CAE tools; and (2) technical discussions with members of the RAMCAD research effort regarding the goals and plans of the RAMCAD Program, the RAMCAD evaluations, and the characteristics of the design process. From this information, a broad range of criteria was created that characterizes the design process and product quality — the most important output of the design process. The following paragraphs briefly discuss the five main categories of criteria created and the measurable criteria in each. The categories do not serve any formal purpose; they are simply used to help organize and conceptualize the criteria. The criteria are defined conceptually (i.e., without specific metrics) versus operationally because this part of the research is generic and an evaluator could use a broad range of metrics for some of the criteria.

Product Quality

The primary focus of almost any new design method or tool is to help the design engineer create products of improved quality. Thus, this is a logical and required category, but it is also one with many dimensions. Each dimension could potentially be the basis for one or more specific

evaluation metrics. Because quality is usually very important to the customer, the metrics in this category are considered more important than those in the other categories.⁷

Seven dimensions of quality have been identified in an attempt to capture the broad range of items needed to determine product quality. Although some of these items are hard to measure accurately without manufacturing the actual product, estimates can be calculated to help determine the overall usefulness of a design methodology or tool.

Functionality. Functionality is a measure of the number and type of different functions or operations a product is intended to or does perform. For example, a fuel pump at the local gasoline station is expected to perform two different functions: supply gasoline at a given rate based on the pressure exerted on the handle, and provide a reading of the amount and cost of the gasoline dispensed.

Performance. Performance is a measure of how well a product performs the functions for which it was built. To continue the above example, the performance of the fuel pump can be based on three measures: its ability to supply the gasoline at the required rate, its ability to accurately measure the amount of gasoline dispensed under a variety of operational conditions, and its ability to accurately state the cost of the gasoline dispensed.

Both functionality and performance are main goals of many CAD/CAE systems that attempt to lower functional and performance design errors prior to product manufacturing. However, it is very difficult to measure these errors because often they are found only after product prototypes are manufactured and tested or comprehensive computer simulations are run.

Reliability. The reliability measurement is used during design in the theoretical sense as a measure of the probability that a product will perform its intended functions for a specified time interval, assuming the operator uses the item within the conditions for which it was designed (Tracy, 1991). For electronic systems designed for the DoD, reliability predictions based on the quality and usage of the components and the environment in which the system operates are widely accepted for determining the reliability of a product during design.

Maintainability. Maintainability measurement during design is defined as the probability that a product will be retained in or restored to a specified condition within a given time when

⁷ Although this point could be argued, one indication of the importance of product quality to the customer is the inclusion of at least some aspects of quality in virtually every contract or specification written. Indeed, even when there is no contract between the customer and producer for designing a product (e.g., a car), guarantees which show the producer's belief in the quality of the product become a major marketing tool.

maintenance is performed by personnel with specified skill levels using prescribed procedures and resources (Tracy, 1991). This is difficult to measure during most design tasks because the criterion depends on more than the design itself. It also depends on the skills, procedures, and resources used during maintenance. However, there are numerous maintainability measures that design engineers and maintainability experts commonly use during the design cycle after making assumptions about these external factors.

Supportability. Supportability is a measure of the resources and infrastructure needed to ensure that a device operates as intended when it is in the field. The effect of new methodologies and tools on the supportability of a design is also hard to determine until the product is operational because assumptions must be made on numerous external factors. However, there are metrics that supportability experts commonly use to predict how changes in a design will affect its supportability.

Manufacturability. Manufacturability is a measure of how readily a design can be produced given specific manufacturing resources and the availability of the materials from which it will be produced. Experts measure the manufacturability of a design through various metrics that include such factors as the cost to manufacture, the materials needed, complexity of the fabrication and assembly processes, and the equipment required for fabrication and assembly.

Life-Cycle Costs. LCC reflects the total cost of designing, manufacturing, operating, supporting, maintaining, and disposing of the product. Although LCC takes into account many aspects of the above criteria, it can be used as a measure of the overall quality of a product when combined with as little as the functionality and performance criteria.

Process Cost

Process cost is the first of three categories aimed at measuring how a methodology or tool changes the design process. A primary goal of many CAD/CAE innovations is to decrease the time and costs associated with the design process while increasing product quality. The process cost is a measure of the costs involved in implementing, using, and maintaining a technology based on labor and tool costs.

Labor Costs. Labor costs are a function of the work force size, productivity, distribution, and skill or knowledge requirements. Work force size and productivity are usually the two metrics most considered when trying to determine whether to use a CAD/CAE system. One objective of CAD/CAE when it was first marketed was to improve the productivity of design engineers and thus lower the number of engineers and draftsmen needed. This, in turn, was expected to reduce

labor costs. However, these work force reductions were not generally observed (Primrose, 1988). Instead, increased productivity means greater throughput by the design engineers, increased product quality, or the need for new personnel to maintain the CAD/CAE systems thereby offsetting the decreased number of engineers and draftsmen.

The impact of this change in the work force distribution instead of the size is an important item when measuring labor costs. Ideally, the change in work force must occur gradually to keep the work force content and productive. Establishing the work force distribution is required to determine that the required number of personnel with each particular skill type are available and that peak staffing will be at an acceptable level.

The labor costs criterion must also take into account the knowledge requirements of the design engineers who will be using the new methodology or tools. New ideas and tools require design engineers to learn new skills so that they can properly use the methodology or tool and interpret the results. However, maintaining personnel skills and knowledge at higher levels can be expensive.

Tool Costs. The equipment and software needed to effectively use a new methodology or tool must be considered. The new methodology or tool may need special equipment such as new plotters, an extra computer screen attached to a workstation, or even new workstations and a change in the office environment. This cost may be large or small based on the methodology or tool.

Process Time

Process time is the second category used to measure the change in the design process caused by a new tool or methodology. This category is important because many people create CAD/CAE innovations hoping to decrease the overall time required to design a product. However, the time required must be measured carefully because, as is often stated in current discussions about IPD, investing more time during the early phases of the design cycle can shorten the overall design cycle and improve the quality of the resulting product.

Process Duration. Process duration is a measure of the time needed to complete part or all of a design. This metric can include measuring the entire design cycle from conceptualization to manufacturing or just take into account individual steps of the design based on what the methodology or tool being tested affects. For example, for the methodology and tools created under this effort, the part of the design process measured could begin late in system engineering, when the methodology and tools are first used, and end during design analysis. Manufacturing, test, and evaluation of the product could be ignored in this example if it is assumed (properly or

improperly) that the tools only provide information about the design that would be normally be obtained prior to the manufacturing, test, and evaluation steps anyway. Thus, the tools do not change the final design of the product; they merely alter when the design engineer obtains and uses information about the design.

However, innovations may affect the interactions among the individual tasks and have intended or unintended consequences. These consequences could then affect the duration of the larger tasks or the overall design process. Measuring these changes may be impossible under real-world conditions. However, care should be taken to ensure that as many consequences as possible are known and accounted for when measuring this criterion.

Iteration Frequency and Duration Distributions. Design iteration refers to the need to repeat some steps in the design cycle one or more times. The iteration frequency distribution is a measure of the number of times design engineers and area experts must repeat certain steps to achieve an acceptable design. The iteration duration distribution is a statistical measure of the amount of time required to complete each step. This distribution can be illustrated by using such statistical measures as mean and median time and by determining the shape of the distribution curve.

Iterations are usually more expensive when performed later in the design cycle. Thus, when measuring the iteration frequency, it is important to understand where in the design cycle the iterations occur. Although a CAD/CAE methodology or tool may increase the total number of iterations required during a design process, it may reduce the total time and cost if the additional iterations required to design a product occur early in the design cycle.

Both measures may be difficult or impractical to determine completely. Nonetheless, rough estimates of these measures should be used along with a confidence level for the estimates to determine how a methodology or tool affects the design cycle.

Training Time. The time design engineers need to learn and become proficient in the use of a new methodology or tool can be an important factor in the success or failure of a new technology. Many organizations may be unwilling to spend sufficient time for the design engineers to become fully knowledgeable. This will cause the design engineers to use the tool inefficiently and ineffectively, dooming the technology to failure.

Training time may be decomposed into the time needed to achieve the basic skills, become a proficient user, and maintain that proficiency. In addition, it may be necessary to include the time required to develop and understand the capabilities of the necessary support technologies. One

example is the idea of design libraries that must be populated before the old designs can be called up for modification into new designs. The benefit of such a technology may not be apparent until these libraries are well populated.

Process Flexibility

Process flexibility is the last category that helps measure the change in the design process. This category measures the tolerance and adaptability of the methodology or tool to a wide variety of design problems. Many of the current tools are limited in use to specific times in the design cycle or to a small subset of all possible design problems. The flexibility of a methodology or tool directly affects its capability to create optimized designs. This statement is based on the belief that if a tool is inflexible, it cannot be used to create or analyze a variety of different designs; thus, the design engineer will not be able to adapt the working environment to a new design problem or to solve a design problem in a way not envisioned by the tool designer. Basically, the tool or methodology, and not the design engineer's capabilities, will limit the design possibilities.

Range of Application. The range of application measures the types of products and range of design tasks for which the tool or methodology is appropriate. A definition of the range of application is essential in determining whether the requirements of a design problem match the capabilities provided by a CAD/CAE system or tool. The risk of a narrow range of applications is that the methodology or tool will not address enough of the problem or that changes in the design problem will render the methodology or tool obsolete.

Design Modifiability. Design modifiability reflects the ease and cost of making changes to a design. A design tool that allows design engineers to easily recover and modify previous designs would have a higher level of design modifiability than one that does not support these capabilities. A lack of design modifiability is one reason many existing design systems are not efficient tools for the design engineer to use during the design process. Because designs are so difficult to modify in many of these systems, design engineers only use the tools to document or verify the results of the design process. Changing a component in a schematic, for example, often involves deleting the original component and replacing it with one which has the desired design attributes instead of simply specifying changes to the attributes of the schematic instance. In this case, the design engineer must often respecify even those component properties that should remain the same. This metric is important because it indicates how usable and efficient the methodology or tool can be during the design process.

Availability of Data. One significant problem with existing design tools and methodologies is that they require a level of detail that prevents them from being used earlier in the design cycle or

to support tasks for which they were not specifically designed. This is a primary reason design engineers use CAD/CAE workstations more to document and verify the design than to carry out the design process. Presently, design engineers prefer to perform many design tasks manually (i.e., with paper and pencil) than to use CAD/CAE tools because of the information required by these tools to perform even the simplest design functions. Ideally the tools should assist the design engineer in reasoning about the design problem from the outset. This is when the design engineer makes crucial decisions about the design problem, with or without computer tools and complete design information.

A related factor is the standards, guidelines, and requirements about which a design engineer must be knowledgeable in order to effectively support the design process. Although the use of standards is not meant to be discouraged, imposing standards or guidelines on the design engineer when they are not well understood or readily accessible significantly affects the design engineers' productivity and, indirectly, their acceptance of a methodology or tool. Thus, any computer-based methodology or tool must include the proper level of on-line help and provide access to the standards and guidelines required by the majority of design problems.

Deferrable Commitments. Deferring a commitment to one approach or another until later in the design cycle, when more information will be available, is one way to ensure the design engineer can design the best possible product to fulfill customer needs. Committing to one set of specific alternatives often precludes the later adoption of a competing approach or alternative that might yield a better design. Also, changes in requirements are less costly if they precede such design commitments. Deferring commitments and maintaining a database of the available options increases the probability that the design will not need modification later.

Design Task Dependencies. The definition of a design methodology should establish the relationship between specific design tasks and each specialist involved in the design. Changing the methodology or tools used in the design process can have a dramatic effect on these relationships and, thus, change when or if specialists should accomplish specific tasks. Although it may be impossible to understand all the relationships required during the design process, anyone designing a new tool or methodology should take care to understand how the relationships will change. Using a new tool or methodology can require so many new relationships that chaos may result or too few relationships may be supported, thereby resulting in a poor design because of insufficient design information-sharing between design tasks or specialists.

Job Satisfaction

This last category takes into account the fact that design is still a human endeavor, even though design engineers use computers to support the process and there is the potential for automating many design tasks through computers. The success or failure of a product inevitably depends upon the capabilities, enthusiasm, and motivation of the design engineers. Design tools and methodologies that do not adequately address human factors are likely to stifle the creativity and motivation of their users, which, in turn, will result in poorer designs and lower productivity.

Skill Growth. Two strategies are used to support complex activities that require skills and knowledge beyond the abilities of the current workers. The first is to develop technologies that allow the computer to perform these activities while humans may or may not monitor the work. This strategy entails replacing human expertise with rule sets and other techniques, thereby requiring a less-skilled work force and, thus, a cheaper labor rate. However, this also deskills⁸ the work force and lowers worker motivation and ability to increase product quality. An alternative is to provide powerful tools intended to aid the skilled worker (Bodker et al., 1988). This strategy and the associated tools allow workers to continually increase their proficiency and achieve higher product quality through human intuition and skill growth. Each strategy has its place based on the work involved and other economic, technical, and political factors. The ideal situation is often one that allows work to progress at a minimum specified quality and productivity level while the workers increase their skills. After sufficient experience, workers should be able to surpass the minimum levels and, eventually, may surpass the capabilities of the tool altogether.

Use of Talent. Most people enjoy using and demonstrating the talents they have acquired. A design methodology and tool that utilizes these talents will be readily accepted and more successful than one that does not. In the worst case, design engineers may reject a methodology or tool that imposes tasks that do not match their skills or areas of interest.

Creative Opportunities. Design engineers generally prefer to perform tasks and use tools that allow them to be creative. A tool or methodology that eliminates creativity demotivates users. However, an organization may prefer and need to limit the amount of creativity a design engineer includes in its products. Thus, one must take into account the amount of creativity allowed by a tool or methodology as well as the amount that the tool or methodology should allow.

⁸ Deskilling can take place under two circumstances: (1) a skilled worker may leave and be replaced by someone with less skill or, (2) due to lack of use, a person may lose the ability to use a skill. While the entire skill may not be lost in the second case, it is possible for workers to become "rusty" and require retraining before they can perform the skill as proficiently as they did originally.

Usability.⁹ User interface is an important dimension of any tool; therefore, the tool designer must consider it from the beginning. As a suite of tools offers more and more information to the design engineer and new tools and methodologies thrust additional complex analyses into the earlier phases of the design cycle, the organization and presentation of design information and analysis results become of paramount importance. Usability is a complex criterion that includes such topics as information presentation, technology/user interaction, and feedback quality. The last topic is of particular importance because providing positive feedback for accomplishments during the design process can motivate the design engineer to new heights in productivity and product quality.

Experiential Qualities. The four criteria listed above contribute to understanding a design engineer's satisfaction with a tool or methodology but they do not measure it. Only the design engineer can evaluate the satisfaction received from using a particular methodology or tool; therefore, to measure this important criterion, an evaluator must directly question the design engineer.

Evaluation Challenges and Approaches

It was anticipated that determining the usefulness of the methodology and tools created under this effort would be difficult. Also, it was anticipated that having numerous design engineers work on different designs while being aided by new tools would create problems for both the design engineers and the evaluators measuring the results of their efforts. To better understand the problems associated with performing this evaluation, a list of the challenges associated with the evaluation was created. The methods and evaluation criteria to be used to measure the outcome were then determined. The results of these efforts are detailed in the following paragraphs.

Evaluation Challenges

During the planning stages of the Phase 3 effort, it was recognized that difficulties would arise in the evaluation of the RAMCAD tools and methodology. These difficulties stemmed from four main sources. First, to perform the evaluation within the time and resource constraints, benchmark problems rather than highly complex design problems would need to be used. Second, the design engineers that participated in the evaluation were unfamiliar with the new methodology, the NS ECAD operating environment, and all the CAD/CAE tools involved in this effort. Third, the new tools were not robust software programs but proof-of-concept implementations meant to convey

⁹ Usability is the user-friendliness or ease-of-use of the tool or methodology.

proposed capabilities instead of employing a full set of capabilities. Fourth, as in any formal evaluation, the design engineers were likely to be swayed into responding based on what they believed the evaluator wanted the results to be, whether or not the evaluator consciously showed interest in the responses. Each of these difficulties are discussed in more detail below.

Benchmark Problems. Measurements taken from benchmark tasks used during an evaluation may differ considerably from measurements taken in an actual work setting. One reason for this difference could be that the evaluation context limits the design engineer's access to other resources, including people and documents, that may be an integral part of the typical design process. Another reason could be that the evaluation process usually removes daily interferences such as telephone calls and requests for advice from other design engineers. These differences could conceivably invalidate the results of the evaluation. In addition, the way the design engineers used the RAMCAD technologies during the evaluation could easily be different from the ways they would use the tools and methodology over a long period on actual design problems. For example, continuously available RAMCAD technologies might enable a design engineer to learn some of the RM&S implications of design decisions. This information would not be obvious during short benchmark tasks because of the design engineer's lack of time and opportunity to learn these implications.

The benchmark tasks were also smaller than the design tasks the RAMCAD technology would normally be expected to address. Different design technologies can facilitate work in either small, simple designs or large, complex designs but not necessarily both. This made it very difficult to determine how useful the methodology and tools would be on larger, more realistic design problems.

Using benchmark tasks limited to only a small section of the design process also limited the ability to determine how the methodology and tools would affect the larger context of the design process. The RAMCAD methodology was created to improve the quality of the design emerging from the section of the design cycle affected by the methodology and tools. This should, in turn, influence the occurrence, duration, and output of later design tasks, especially during the RM&S analysis and improvement tasks. If the methodology and tools worked as expected, any results received would not fully reflect the benefits of the methodology and tools.

Unfamiliar Operating Environment. The participants in the evaluation were required to learn to use different and new technologies. The design engineers who participated in this evaluation had not used a Symbolics machine, the NS ECAD design environment, or the tools created under this research effort prior to the evaluation. Because their design work took place

while they were on the beginning of the learning curve for these technologies (where the slope is steepest), it is likely that the measurements underestimated the performance achievable with this technology. The evaluation data were also expected to exhibit large design engineer performance changes within the evaluation sessions which would be hard to quantify during subsequent analyses. An attempt to alleviate this problem was made. The design engineers worked on two practice designs before beginning the benchmark tasks. Unfortunately, it was impossible to determine how much success this approach had in alleviating the problem.

Insufficiently Robust Tools. The RAMCAD methodology and tools were proof-of-concept versions that could not possibly achieve the full benefits expected from commercial quality versions. That, tied with a user interface that had not been tested for usability, forced the quantitative measures to be below what is possible with a full set of robust tools. Consequently, the benchmark tasks were used more to assess the potential benefits provided by the methodology and techniques than to assess the tools themselves.

Experimenter Bias. The design engineers knew that the new tools were expected to improve the RM&S of the design; therefore, they were more attentive than normal to these factors while working on the design problems. Improvements that were seemingly wrought by the tools and methodology may be due, in part, to this effect. An attempt to remove or reduce this bias was made by emphasizing the functionality, performance, and RM&S characteristics of all the design problems, whether or not the designers could use the tools. The participants did realize, however, that RM&S was expected to increase with tool use; thus, they probably tried harder, believing that this was the favored condition. This problem was unavoidable, but the effects were expected to be negligible because many design engineers have a limited knowledge of most of the complex techniques used to improve the RM&S characteristics of a design.

Evaluation Approaches

Two different approaches were used for the evaluation because of the differences in the nature of the tools, the capabilities included in them during this effort, and the available resources. The first approach described below was used to evaluate SIDECAR and STA; the second approach was used to evaluate ITA. ScanIt was not evaluated under this research effort.

SIDECAR and STA Evaluation Approach. The approach used to evaluate SIDECAR and STA was meant to determine their capabilities at both subsystem design and detailed design levels. It included using design engineers solving small benchmark problems both with and without the RAMCAD tools. The results of the design engineers' work were then analyzed to determine whether the tools assisted the design engineers in creating improved designs.

Boeing avionics subsystem engineers and detailed designers were tasked with creating four problems from each of the two design levels for this evaluation. Each problem was required to be solvable in a single day, yet with performance, cost, and area criteria stringent enough to make solving the problem a challenge.

In parallel with the problem creation effort was an effort to create an electronic parts library for the new tools. It required sufficient information to allow numerous possible solutions to be found for each design problem. Building of the library began through requests to the design engineers who created the problems to indicate those items that should be included in the library. The items had to provide reasonable choices for a design engineer to make while creating designs to satisfy the design problems. Data associated with these parts was entered based on whether the item was needed for a detailed design or a system design problem. For detailed design, parts data was obtained from various data books. The data included such information as the gate counts and power dissipation figures required to compute reliability estimates as well as the burdens in area and cost associated with each part. For system design, components were added to the database and included such parameters as cost, reliability, and Pfd for each.

The fact that the design engineers normally had little or no notion of the cost and reliability of various parts created a problem. If this information was not supplied to design engineers not using the new tools, they would have to rely on their best guesses as estimates. In "real life," this data is available from specialists, although it can take quite some time for the specialists to respond to requests for information. The solution used for this evaluation was to provide a hard copy of the information contained in the database. This made it easier for the design engineers to use this data than in "real life," but it seemed to be the best compromise.

Once the electronic parts library was created, the design problems were calibrated. Two system engineers and two detailed designers were used for this task. Each spent a day learning the NS ECAD workstation and all the tools associated with the evaluation, then worked four days solving the problems in their area (i.e., one problem per day). This confirmed that the problems would each require approximately one day to solve. In addition, the problems and their goals were modified to reduce ambiguities and ensure they were properly challenging. One major problem during the calibration exercise was that the design engineers tended to ignore the testability requirements because they were unfamiliar with the testability metrics used to express the requirements. In addition, since there were no constraints on the connectors, they felt they could meet any testability requirement by running a wire to an unused connector wherever necessary. Unfortunately, all attempts to solve this problem were unsuccessful and it caused the evaluation effort to be less than completely successful.

After changing the problems to compensate for the deficiencies noted above, the evaluation was performed using two system engineers and two detailed designers. Each design engineer spent a day learning the NS ECAD workstation and the other tools, then spent four days creating designs to satisfy the design problems. On any given day, the two system engineers worked the same problem, as did the detailed designers. In each case, one used the RAMCAD tools and one did not, with the use alternated between the two daily.

The design engineers were asked to record the time required to solve the problems. The intention was to normalize these times to compensate for differences in the speed between design engineers, then compare them to determine the time required to complete each problem with and without the RAMCAD tools. Unfortunately, the design engineers failed to properly record these times. The design engineers were also required to complete a questionnaire after the evaluation to allow their reactions to the RAMCAD tools to be determined.

ITA Evaluation Approach. A different approach was used to evaluate ITA because the ITA analyses depend on models of the design components. Creating the required models for all the components was beyond the resources allocated for this research effort. Furthermore, ITA has a tendency to err on the side of optimism in analyzing for testability. Characterizing the nature of this optimism was a very important aspect of the tool evaluation. The approach selected for this effort required a testability expert who understood the tool. In addition, a fault simulator was used to test ITA accuracy. Because the fault simulator required models and test vectors from completed designs, the only feasible approach was to test ITA on complete designs.

Three real electronic designs with differing gate-counts (8, 54, and 209 gates) and component complexity were selected for this effort. Because of the design stage used to evaluate ITA and the tool capabilities available during the evaluation, only the accuracy and execution speed of ITA were evaluated.

Developing Evaluation Criteria and Methods and Evaluating and Validating the Research

A different set of evaluation metrics was used for each evaluation approach. This section describes each set based on the tools it measured; metrics are grouped where appropriate. In addition, the evaluation and validation results are presented.

Although an attempt was made to evaluate the tools and validate the methodology through the benchmark design problems, several quantitative aspects of the experiment were inconclusive. Thus, most of the results discussed below are more qualitative than quantitative. Still, the

evaluation results were largely positive and the design engineers indicated they would like access to tools that performed the functions demonstrated by SIDECAR and STA. They also believe that the methodology required to use these tools would have positive influences on design quality while not adversely affecting the total design cycle time requirements. Furthermore, ITA proved to be acceptably accurate and efficient.

System for the Interactive Design and Computer Analysis of Reliability and Statistical Testability Analyzer Metrics and Evaluation

The SIDECAR and STA metrics are a subset of the metrics described earlier in this report. Not all the evaluation metrics discussed above were employed in the evaluation because of the design methodology used, tool limitations, and the size and scope of the evaluation. The following paragraphs present the metrics used to measure the data, the data-collection method, and the results of this effort.

Product Quality

FUNCTIONALITY. The goal of this research effort was to improve the ability of design engineers to address RM&S concerns early in the design process. Although the tools did not specifically analyze functionality, each design engineer was expected to satisfy the functionality requirements while improving the reliability, availability, and maintainability of the design. This expectation, and the fact that functionality is a qualitative criterion, led to the use of area experts to compare the level of functionality of the designs created under this evaluation with those expected from the current design process. The intent was for the design engineers to increase the RM&S aspects of the design while at least maintaining similar levels of functionality. Qualitative analyses were used to determine if the designs created with the new tools met or exceeded the functionality requirements and how they compared to the designs created without the new tools. Either way, it was decided that little significance should be attached to the findings of this analysis other than ensuring the design met the functional requirements.

During the calibration experiments, the design engineers spent most of their time drawing traces to connect circuit components. This time-consuming task used up most of the time required for design analysis. To add more time for analysis, the design engineers were encouraged to use the RAMCAD tools before they specified connectivity as long as they were certain the circuit would work after they specified the connectivity. The design engineers were also told that component selection and design improvement techniques were more important than the details of connectivity. Consequently, the design engineers spent very little time specifying the connectivity of the components during the experiments. As a result, the functionality experts were unable to

perform any meaningful analysis on the designs. All that could be determined from observations and post-design interviews was that the design engineers took the functionality requirements very seriously and would probably meet the criteria if given the time to complete the circuit connections.

PERFORMANCE. Performance, like functionality, was not meant to be analyzed with the RAMCAD tools during the evaluation. Furthermore, performance can be both a qualitative and quantitative criterion. Both aspects of this criterion were used because limiting it to a quantitative criterion would require too much detailed parts information and completed design connectivity than time allowed. Quantitatively, throughput or device speed for digital devices and precision and signal-to-noise ratios for analog devices were used as metrics. Qualitatively, area experts were used to compare the performance of the designs created under this evaluation with those created under the current design process. Either way, it was decided that little significance should be attached to the findings of this analysis area.

Rating the performance of the designs fell into the same trap as functionality: the experts could not measure either the quantitative or qualitative criterion because of a lack of component connectivity. However, the design engineers took performance requirements very seriously and felt they had met the criteria required by the problem statements.

RELIABILITY. Quantitative reliability requirements were placed on each design problem so they could be compared directly in addition to a subjective analysis of the designs. For the system design problems, the reliability metrics were mission completion success probability (MCSP) and mean time to failure (MTTF). For the detailed design problems, the metrics were MTTF, MTBF, and mission time. Reliability engineers performed the subjective analysis to determine the effect of using the RAMCAD tools on the design problems.

The system design results were inconclusive. Only for one problem did either design engineer fail to meet a requirement; for this problem, both failed to meet it. The design engineer using the RAMCAD tools developed a system that was significantly less expensive but only obtained 60 percent of the required MTTF. The design produced without the RAMCAD tools obtained 95 percent of the required MTTF. In the remaining cases, the design engineer always produced a significantly less expensive design when using the RAMCAD tools. However, there was no clear trend on the reliability metrics.

In the detailed design experiments, the numbers were also inconclusive. One detailed designer always had a less expensive design and in three of the four design problems had better reliability metrics (in the fourth problem the detailed designers had similar reliability results). In only one case did a design not meet a requirement. In that case, the detailed designer had used the

RAMCAD tools and produced a design with less than four percent of the required mission time. In all other cases, both detailed designers met all the goals with no consistent differences between the designs produced with the tools and those produced without them.

Several factors contributed to the similarities and inconsistencies in the experiments. First, on the first day the approach to computing various reliability measures was reviewed while the design engineers were taught to use the NS ECAD workstations. Second, the design engineers had access to a reliability expert throughout the experiments who could answer their questions on proper reliability computing techniques. These two factors provided the design engineers information that was not common knowledge and aided their effectiveness. Lastly, these benchmark problems were small and made hand calculation easy. Since the design engineers had access to spreadsheet programs throughout the exercises, it was easy for them to perform the reliability analyses with only a small time penalty if they did not have access to the RAMCAD tools. This would not be the case for realistically complex designs created under normal working conditions.

MAINTAINABILITY. Quantitative maintainability requirements were compared for each design problem and area experts subjectively analyzed the designs. Because the focus was on the testability aspect of circuit design, each design problem contained quantitative testability requirements expressed as a minimum required Pfd and minimum required probability of isolation of a detected fault to one, two, or three components. Furthermore, the maintainability experts analyzed the designs for such measures as frequency of repair, operational level MTTR, maintenance man-hours per flight hour, and fault detection and isolation time.

Unfortunately, the system engineers were very inconsistent in the amount of attention spent on the testability goals. Because so little information was provided about the design, they believed they could assert virtually anything about testability; this meant there was nothing meaningful for them to do. The detailed designers were also inconsistent because of the lack of connector limits. They thought this lack meant they could tie every pin to a connector, thereby providing direct access for test and meeting all requirements. Unfortunately, this prevented any meaningful analyses of the designs for testability.

SUPPORTABILITY. The simplicity of the design problems largely removed the significance of supportability metrics. However, a supportability expert was tasked to calculate rough estimates of quantitative supportability measures such as logistics support cost (LSC), spares availability, and field support equipment needs. Unfortunately, the calculations, which would have helped identify any improvement in this area, were impossible to perform because of the lack of connectivity.

MANUFACTURABILITY AND LCC. The simplicity of the design problems also removed any significance the manufacturability or LCC criteria could offer. The design engineers were not expected to address manufacturability or LCC concerns, and the RAMCAD tools do not address these areas.

Post-evaluation interviews were held with the design engineers to determine the impact the RAMCAD tools could have on product quality criteria. During the interviews, the design engineers were asked to estimate the effect robust versions of the RAMCAD tools used in a familiar CAD/CAE environment would have on the design cycle. They responded that, although not all these criteria were measured, they believed robust versions of the RAMCAD tools could measure all these areas. They also believed the CAD parts library could act as a preferred parts list and, if these factors were taken into account when selecting parts for the library, design engineers would be forced to involuntarily account for these factors also. Additionally, if quantitative metrics for the manufacturability and supportability criteria were added to the library, design engineers could use a SIDECAR-like tool to create designs that directly took these factors into account.

The design engineers also believed that a combination of SIDECAR-like tools and an LCC model would effectively support LCC analysis. If properly created, the design environment could automatically extract most of the data needed for the LCC analysis from the design and parts library. The system engineers could provide any other information required to do a proper LCC analysis at the same time requirements are sent to the lower-level design engineers. If they made any changes to the LCC analysis information, they could pass this information on just as changes in the requirements are passed on.

Process Cost. The evaluation of process cost was primarily qualitative and based on the interviews with the design engineers participating in the evaluation. The objective was to determine whether the RAMCAD tools would have a significant positive or negative effect on the process cost evaluation criteria.

LABOR COSTS. The approach was to base the labor costs mainly on the size of the work force needed and the productivity of the design engineers. These, in turn, would be based on whether there was a large difference in the solutions to the design problems and whether fewer reliability and testability specialists would be needed during actual design work and post-design analysis and modification. Additionally, a qualitative analysis was used to determine if the proposed techniques and tools would require the design engineer to acquire vast amounts of additional design and computer skills or if the RAMCAD tools provided an acceptable level of specialist help.

The inconclusiveness of the quantitative aspects of the reliability and testability analyses prevented a quantitative determination of the change in work force size. In general, the design engineers believed the tools would not reduce the number of design engineers or specialists required during the design process. They did agree that the tools would lessen their dependency on reliability engineers. Unfortunately, they also thought this would not have a significant impact on the total number of specialists required. Thus, no changes would be expected for this criterion.

TOOL COSTS. The costs associated with using the tools was qualitatively analyzed. However, the analysis was limited to determining whether a standard design environment would need special equipment to operate the tools. It did not include the costs of creating, buying, or maintaining software. Based on the work required to create and evaluate the proof-of-concept tools, software vendors should be able to create commercial versions of the tools to run on the systems currently in use in industry. Design engineers would not need special equipment to operate these tools.

Process Time. A qualitative analysis was performed to evaluate the impact RAMCAD tools would have on process time. Furthermore, the design engineers were asked whether they thought the tools would have an impact on the various criteria associated with this category. They were also asked to determine the amount of training the average design engineer would need to use the tools effectively.

PROCESS DURATION. The process duration analysis was limited to determining whether the time required to perform design tasks would be reduced when the tools were used as anticipated. This measurement was to be quantitative and based on a normalization of the times required to perform the tasks. The design engineers were asked to record the time required to perform various aspects of their work for this express purpose. The proposed approach was to normalize the data for each design engineer, then compare the data points to determine if there were any clear changes in the process durations associated with the use of RAMCAD tools. In addition, an ability to determine the number of times during the design tasks the design engineer would need information from specialists, the time needed by the specialist to provide the information requested, and the number of times the tools could satisfy the request was anticipated. Unfortunately, the design engineers failed to accurately record the time data on a consistent basis, so a quantitative analysis to determine if there were any changes in the process duration could not be made.

Questions about the process duration were included in the post-evaluation interviews with the design engineers in an attempt to gain some data on this criterion. The design engineers responded that the RAMCAD tools would lessen the time required to complete a design if they were required

to optimize an actual design problem for more than just functionality and performance. However, currently they are not required to perform this function.

The system engineers perceived a large impact in their area. They believed the tools would facilitate trade studies among more factors more quickly than current tools and processes. In addition, the immediate feedback from the tools should lower the daily dependence on specialists; ideally this would reduce the demands on the specialists and allow them to spend more time helping all design engineers on more difficult design problems.

The detailed designers expected less of an impact in their area, primarily because they spend little time analyzing their designs for reliability or testability. Instead, most detailed designers believe (usually correctly) that if they stick to the preferred parts list and the fault tolerant techniques specified in the design requirements, they will meet the reliability and testability goals.

ITERATION FREQUENCY AND DURATION DISTRIBUTIONS. The iteration frequency and duration distributions criteria were measured based on the number of iterations observed for the tasks that used the new design tools and methodology. The measurement was also based on whether the design engineers improved the designs sufficiently to lessen the number of times they must modify the designs to meet criteria changed downstream in the design process.

Quantitatively, these criteria could not be measured for the same reason the functionality and performance criteria could not be measured. However, qualitatively, the design engineers agreed that access to the RAMCAD tools, coupled with requirements that made it necessary to consider the aspects addressed by the tools, generally increased the number of design iterations and the speed of each iteration. Overall, they did not think there would be a significant impact on design time, although the most likely effect would be a decrease. They also agreed that the quality of the resulting design would increase because of an increased ability to perform trade studies and optimize the design.

TRAINING TIME. The change in training time for design engineers was estimated based on the design engineers' answers to questions during the interviews and an estimate of the additional skills needed to employ the proposed tools and methodology. The design engineers disagreed on the amount of training time required to use the RAMCAD tools. They stated that the actual use of the tools was simple and straightforward but to use them properly and understand the results requires significant training in the RM&S areas. The design engineers' perceptions of the required training time seemed to reflect their varying expectations of the required depth of knowledge in the RM&S areas and what the tools could perform if they were commercial-quality tools instead of proof-of-concept versions.

Process Flexibility. There were no quantitative metrics associated with the process flexibility criteria; thus, measurement was purely qualitative. All conclusions for the criteria in this category (i.e., range of applications, design modifiability, availability of data, deferrable commitments, and design task dependencies) are based solely on the results of the interviews with design engineers. These interviews addressed the ease with which they could make design modifications, the interdependencies of the data and individual design techniques, and whether the techniques would enable the design engineers to make better design decisions.

The design engineers believed the RAMCAD tools would be useful in creating any design for which reliability and testability are key issues. In general, they thought the tools would be most useful during concept creation, less useful in system design, and least useful in detailed design. Within each of these areas, the design engineers expected the tools to work best in the preliminary stages in which they perform most trade studies (although this was not universally accepted).

Job Satisfaction. There are no quantitative metrics associated with the different criteria in the job satisfaction category. Thus, measurement in this category was also purely qualitative. In addition, because the tools are proof-of-concept-level creations and are not robust enough or designed to be easily implemented, the usability of the tools was considered relatively unimportant. The analysis in this category was based solely on interviews with the design engineers. These interviews addressed such items as skill growth, creative opportunities, and use of talent.

The design engineers agreed they would use the tools if robust versions were available in a familiar CAD/CAE system. Overall, they liked having access to additional information about their designs and believed the tools would help them create better designs even if they were not directed to specifically consider the RM&S areas. There was complete agreement that the ability to consider design quality from new perspectives while making design decisions would enable them to produce better designs and, in turn, make the design task more rewarding.

Inherent Testability Analyzer Evaluation Metrics and Evaluation

Because of the different approach to evaluating ITA, only three criteria were used to determine the relative worth of the tool techniques and the tool itself: accuracy, execution speed, and range of applicability. These metrics and the results of the evaluation are discussed in the following paragraphs.

Accuracy. The accuracy of the ITA tool is not as good as originally anticipated because of the controllability function of the tool. ITA accepts four possibilities for controllability of a digital device; a pin may be (1) not controllable, (2) controllable to only zero, (3) controllable to only one,

or (4) controllable to either zero or one. In the last case, ITA analyzes both options (pin set to zero and one) simultaneously instead of individually during the analysis. The optimistic analysis results come from this simultaneous analysis in which, during any individual test, the tool allows the pin to be both values. Consequently, ITA overestimates the testability of many redundant and reconvergent circuits.

The purpose of the accuracy evaluation was to quantify the expected incidences of the error and to characterize the faults involved. After each ITA analysis, a subset of the faults identified by ITA as testable were selected for fault simulation analysis. Only subsets were used because of the scope of the evaluation and the sheer number of testable faults that would need analysis if all of them were chosen. After analyzing the faults, an attempt was made to identify and classify them, and to statistically classify the population.

The evaluation of ITA accuracy was based on experiments with two different circuits, a full adder (8 gates) and a 16-bit carry-look-ahead circuit (54 gates). In these experiments, the results of ITA analyses were compared with the results of a fault simulator, which were taken as the baseline for accuracy comparisons.

For the full adder, there were three input pins and two output pins. Since each input pin can have four possible controllability values and each output pin can have two possible observability values, 256 combinations were possible for the test. However, these were narrowed to 192 for the experiment by ignoring any combination that included values of no observability for both output pins. The fault simulator identified a total of 54 possible stuck-pin faults for the adder. In 126 of the 192 combinations (66 percent), the results of the ITA analysis were identical to those obtained through fault simulation. Only in some of the cases, for which at least one pin was specified as controllable to both 0 and 1, did ITA err. With only one exception, assigning controllabilities that caused ITA to give an error led to erroneous results for all three observability combinations. When an error occurred, the average estimate of testability by ITA was 151 percent of the true value. The extremes included 12 cases for which ITA identified 29 faults as testable when only 28 were actually testable (104 percent) and three cases for which ITA identified 26 faults as testable when none of them were testable. In 75 percent of the cases, the ITA estimate was less than 110 percent of the true value. In 81 percent of the cases, the ITA estimate was less than 120 percent of the true value.

A likely use of ITA is to evaluate a single set of input controllability and output observability values for a given circuit. ITA is optimistic when it errs; this is a positive feature because when ITA identifies a component as having poor testability, the component is a good candidate to

consider for testability improvement efforts. On the negative side, ITA may not properly identify the component with the poorest testability since it could overestimate component testability. Furthermore, if the single evaluation being performed happens to be on the tail of the curve and, as in the adder circuit, shows 26 of 54 faults as testable when none are, the analysis results would be significantly misleading.

ITA scattered the instances somewhat when it made errors in its estimates and, thus, no generalizations could be made. For instance, pin controllability consisted of two pins set for complete controllability (i.e., controllable to both 0 and 1) while the third was not controllable in the case of 26 faults being considered testable when none were. However, when the third input was modified to be controllable to 1, ITA identified 44 faults as testable when 38 were really testable; this was a significant improvement. If the third input was modified to be controllable to either 0 or both 0 and 1, ITA analysis results were completely accurate. These results made it impossible to generalize when ITA would give misleading results and how inaccurate these results would be.

The 54-gate circuit was only tested for one set of plausible inputs. This was a high fan-in circuit that had 32 inputs and 1 output leading to 1.845×10^{19} possible input/output combinations. In the one test, both ITA and the fault simulator identified 70 faults as testable. Only one test was run because the fault simulator required several thousand steps of fault simulation to evaluate each test of this circuit.

Execution Speed. The execution speed was evaluated by measuring the speed required to analyze three test designs having known device counts. The three metrics to be determined through this method were the time required to perform an analysis of a standard size circuit in a production implementation of the tool, the correlation between evaluation time and circuit size, and whether the ITA analysis algorithm is efficient enough for a design engineer to use interactively during design or only as a post-design analysis tool.

The Symbolics model 3640 computer on which the experiments were performed is not a high-performance machine by today's standards. Also, ITA was not coded for execution time efficiency. Thus, it is assumed that using a high-performance machine with an optimized code would increase tool performance by approximately one order of magnitude.

The speed of ITA was estimated by measuring its run times on three circuits — the two mentioned above and a 209-gate circuit. The current implementation of ITA is less efficient on hierarchical circuits than on flat circuits. (The adder is a flat circuit while the other two are

hierarchical.) Thus, a direct relationship among the results from these three circuits should not be considered accurate.

The time ITA used to analyze each of the 192 combinations of controllability and observability on the adder was measured. The average time was 0.83 seconds. Two runs for the 54-gate circuit were timed, each with different combinations of nearly complete input controllability. The run times were 4.03 and 5.03 seconds. Three evaluations of the 209-gate circuit were timed: one with no controllability to estimate minimum run time, one with total controllability to estimate maximum run time, and one with nearly complete controllability representing a typical case. The run times were 17.6, 68.5, and 30.0 seconds, respectively.

The average run time per gate was 0.104 seconds for the adder, 0.0839 seconds for the 54-gate circuit, and 0.185 seconds for the 209-gate circuit. The time for the typical assignment of controllability on the last circuit is 0.144 seconds per gate. In this limited experiment, the increase in processing time to increase in circuit size was nearly linear. Accurate determination of this relationship would require more extensive tests on a wider variety of circuit sizes, topologies, and complexities.

Range of Applicability. The current implementation of ITA is limited to digital designs. Simple combinational and sequential circuits were evaluated with equal ease. More extensive experimentation might reveal limitations of its application based on circuit topology. There is no known reason why ITA would not be applicable to more complex designs if the required models were available. However, to have any confidence in the accuracy of the tool in such situations, a significant amount of additional experimentation must be accomplished.

General Discussion and Conclusion

Based on the evaluation results, it appears that robust versions of the RAMCAD tools would be effective additions to a CAD/CAE environment. SIDECAR and a parts library would provide design engineers with easy access to information they either do not have in the present environment or must spend a large amount of time to obtain. Most design engineers were enthusiastic about the possibility of having easy access to this information and believed they would produce better designs by using the information with the proper tools. However, this enthusiasm was in part predicated on using the design methodology that forces certain requirements for the attributes measured by the RAMCAD tools. The current design methodologies do not bring these requirements to the design engineer's attention.

Although STA received little real evaluation because of deficiencies in its implementation and poorly designed evaluation problems, the interviews associated with its evaluation demonstrated that STA provides the design engineers access to information not currently available. Thus, it would enable them to optimize designs for various testability attributes largely ignored under the current design practices.

An independent evaluation of ITA could not be provided; however, this tool was developed and evaluated by a testability design expert and it satisfies the needs he identified in his design work. The evaluation validated the concept for at least relatively simple circuits. There is every reason to believe that ITA would be very effective if it were modified to accomplish the following two objectives. First, ITA must be changed to solve the problem it has with performing analysis of circuits with inputs controllable to both 0 and 1. Second, ITA must be available to design engineers in a wide variety of CAD/CAE environments.

V. LESSONS LEARNED

This section includes the more substantial problems that were confronted while performing this research effort. These problems are grouped under the following headings: optimization; design data, knowledge, and expertise; similarity analysis/evaluation; evaluator tools by themselves are not sufficient; rules, heuristics, and guidelines rationale; computer support for design; and design representation. Each group is presented in a subsection which includes a general discussion of the problems confronted and the solutions used (if the problem was solved during this effort) or any proposed solutions expected to help others performing future research in this area.

Optimization

The initial plan was to investigate approaches to automating the optimization design process for RM&S. As part of this effort, research and other support was provided for the development of a design adviser to synthesize reliable designs in the domain of single-board computers.¹⁰ Although the design adviser worked well in the domain of commercial, single-board computers, this optimization approach was unacceptable to the Boeing design community. The design engineers stated that such an automated optimization approach cannot address enough of the design problems encountered in complex military systems to be practical. (They estimated that such an approach

¹⁰ This work was part of an ongoing effort at CMU. For this research effort, CMU personnel created the Automated Synthesis of Reliable Systems (ASSURE) program. ASSURE designs single-board computers in conjunction with another CMU computer program (Microprocessor Configurer [MICON]).

would only be applicable to, at most, 20 percent of the typical design situations.) Additionally, the effort required to convert the design adviser from an autonomous tool to one that allows the design engineer to take an active role in the design process was prohibitive. Instead, it was determined that developing computing aids to assist design engineers in optimizing the design rather than attempting to develop methods that would automate design synthesis would be more fruitful.

In attempting to develop design advisers that would aid design engineers during the early parts of the design cycle, it was determined that the rapid change in electronics (e.g., integrated circuit, component, packaging, and manufacturing technology) and CAD/CAE technology limits the design engineers' ability to develop a design optimized for a variety of requirements/objectives. The amount and direction of the changes in these technologies are largely unpredictable within the life of most military programs (because of the long development periods of the programs). Consequently, the process of predicting which technologies will be sufficiently mature and in widespread use (i.e., adequate manufacturing capability, operational experience, and individuals skilled in their use and maintenance) is a major source of error and uncertainty in the early design decision-making process. The economics associated with alternative technologies, design implementations, and manufacturing techniques change so dramatically during the development program that the most cost-effective approach today will not necessarily be the most cost-effective solution tomorrow.

For example, the B-1 bomber electronics design engineers in the early 1980s could not foresee the current, market-driven revolution in digital electronics. The technologies a design engineer would select today would be significantly different from those chosen by the original design engineers.

Optimization is further complicated by the fact that the design areas which are the greatest contributors to the unreliability and untestability of any specific design are not necessarily amenable to being improved. Furthermore, those areas that the design engineer usually can improve are not necessarily areas that would show the greatest improvement in overall reliability and testability metrics. Thus, a single measure of merit or effectiveness is not the most appropriate means to determine whether a design is optimized. The optimization process must ensure that all areas meet minimum design specifications and adhere to given design guidelines and rules. The process must also facilitate an exploration of the design space to determine where the design engineer's efforts can achieve the greatest design improvements.

A survey of the data collected to date by Boeing Commercial Airplanes indicates that most faults are caused by errors in board assembly (e.g., bent pins and stray washers) and by the faulty

production of a particular part by a particular supplier. This indicates that, at the design level, simplifying the assembly process would be one of the most effective steps to improve reliability. Note that Military Handbook (MIL-HDBK) 217E does not directly address either of these causes.

Two other problems with building an entirely "automated" optimization process are the scope and cost of the system. First, the scope of the system would be much too large because it would require an automated synthesis capability at the system level. This is beyond both today's technology and understanding of the design space. Second, the effort required to develop and maintain an automated tool must be weighed against the productivity and quality gains provided by the tool. Because both the electronics and CAD/CAE technologies are changing at a rapid rate, decision support aids (tools that aid the design engineer in creating and optimizing the design) are more cost effective than automated synthesis tools.

Design Data, Knowledge, and Expertise

One major problem encountered while performing this research effort was the lack of design data, knowledge, and expertise for early design analyses. In many cases, the readily available and published information was inadequate for this effort. Furthermore, reliability, testability, and other design analysis fields had not developed the data, knowledge, and expertise needed to empower a more effective concurrent design process. Consequently, many of the specialists who were interviewed were not prepared to make specific proposals about how they could take a more active role in the early phases of the design process. For example, few testability prediction and analysis methods were found that the design engineer could apply early in the design process. Similarly, manufacturing design has little to contribute to the design process besides basic cost information and general guidelines until the design engineer defines a physical design.

Additionally, many individuals in the engineering community are unaware that most current analysis methods will not effectively support a more concurrent design process. Since many of these analysis methods have evolved to support a sequential design process, design engineers and analysts cannot use them with accuracy concurrently or earlier in the design process without modification. To enable a concurrent engineering process, researchers must develop methods and models to accurately predict the design characteristics of high-level and incomplete designs. Also, they must decompose current analysis tasks into a finer-grained set of subtasks that design engineers and analysts can interweave to produce a more concurrent design process. Unfortunately, the basic information needed to develop an understanding of what does and does not work, the reason for that result, and the models that relate key design features and properties of partial and incomplete designs to the result are lacking.

A second problem in this area is that the current process of satisfying RM&S requirements has become an exercise in data generation versus design improvement. Many RM&S specialists merely verify and document that a design meets RM&S requirements. If the design does not meet the requirements, they can recommend changes; otherwise, they do not make contributions strictly to improve the design. As a consequence of this and constrained budgets, RM&S engineers are not as integrally involved in the design process as they should be. Therefore, they do not have the opportunity to develop new techniques or gain the expertise needed to effectively interact with the design engineers. Many of these "ility" specialists are also reluctant to suggest changes to partial or high-level designs because they have limited experience working at that level. Their experience is typically in the detailed and completed design realm. Combined, these factors demonstrate that the current design process has not prepared the "ility" specialists to work in today's IPD environment.

A third problem in this area is that nobody can properly implement an IPD process, nor improve the design process, without requiring that design engineers have additional design information. This extra data, not currently acquired, is needed to perform the analyses earlier in the design cycle and requires the design engineer to perform additional tasks. However, collecting this data will be very difficult in many design organizations. For instance, design, manufacturing, and field data are often not available or, if available, are in a form that design engineers and analysts cannot use quickly to support the design process. In addition, the available information is insufficient to develop the level of sophistication and quality of prediction/estimation needed to optimize the design.

These three problems required more effort to be placed into problem definition, concept development and evaluation, and basic information gathering than was originally anticipated; yet, this research effort only scratched the surface in solving these problems. To ensure the current thrust towards IPD is successful, researchers must solve these problems and implement solutions that take into account more details than this effort could using simple proof-of-concept tools.

Similarity Analysis/Evaluation

Early in the design process, design engineers and analysts derive estimates and predictions of the characteristics of a proposed design (e.g., reliability and cost) from the properties of similar designs via a similarity or comparative analysis. At present, however, there are no detailed standard methods specifying how to select similar systems or how to adjust the properties of the similar systems to reflect differences (e.g., technology differences and operating environment differences) among them. Consequently, similarity analyses are not automated and adjustment

methods tend to be relatively crude. This, in turn, causes the quality of the estimates to vary with each design engineer. There is no assurance, for example, that the design engineer appropriately accounted for all significant differences between the designs, such as packaging technology or maintenance concept.

Collecting the necessary data to perform a similarity analysis is a large task, complicated by the difficulty of obtaining accurate field data on the RM&S of operational systems. A two-tiered maintenance approach might help mitigate this problem to some extent, since the required data can be collected at the repair depots, thereby lessening the need to depend on the accuracy of data provided by personnel in the field. A related problem is acquiring design and manufacturing information (e.g., cost and manufacturing details) to aid in predicting these factors. The companies who originally build a system often consider this proprietary information.

A second problem is an open design loop. Presently, there is no effective mechanism for providing field experience to design engineers because of the structure and length of military design programs. Although Boeing, like other large companies, attempts to take advantage of government, manufacturing, and internal experience analysis centers, the process of gleaning design lessons from these sources is often so lengthy that the centers cannot provide the information to the design community in a timely manner. Rapid changes in technology and the dispersal of design teams, possibly to different companies, at the completion of the design effort further compound this problem. In many cases, the design engineers of a system never learn how their designs performed in the field.

Evaluator Tools By Themselves Are Not Sufficient

The evaluation performed during this effort showed that simulation and evaluation tools, such as the RAMCAD proof-of-concept tools, are not sufficient by themselves to ensure that the design engineer produces the best possible design. The tools suggest areas in which design engineers can make design changes to improve a given design and rank potential changes according to their impact on the overall design. Unfortunately, the design engineers involved in this effort did not efficiently employ the tools because they did not know how to design for the RM&S criteria they were asked to satisfy. They attempted to optimize the design by proposing a large number of design alternatives and having the tools evaluate each. Essentially, they performed a "random walk" through the design space.

This random walk approach was possible because the tools enabled them to quickly evaluate proposed designs and identify changes to improve them. However, they lacked the knowledge

needed to develop a plan or strategy for meeting the combined design objectives. This, in turn, meant they could not efficiently design for the combined criteria (e.g., reliability and testability) and synergistically employ the analyses and recommendations of the tools. For example, instead of proposing a redundant architecture as an initial design for a problem, they would propose a nonredundant one and be guided to the former by the evaluation and recommendations of the tools. The preferred approach is to develop an overall design strategy for achieving the defined goals and to use this strategy to propose an initial set of alternatives to evaluate and guide the exploration of the design space.

This indicates that researchers must improve their knowledge of how to design for criteria typically specified in practice, train design engineers so they are knowledgeable of alternative strategies for achieving these criteria, and develop tools that are capable of providing guidance on design strategies, such as the proof-of-concept tools created during this research effort.

Rules, Heuristics, and Guidelines Rationale

As part of the research effort, RH&Gs were gathered that the proof-of-concept tools could use to aid design engineers in performing design analysis. This information could be employed in the design analysis through intelligent tools. However, rapid changes in current technology can quickly make this information outdated. Thus, in any future attempts to use this information the tool designers must record the rationale for each RH&G. This will allow design engineers to determine the reasonableness of each RH&G for each new technology.

Computer Support for Design

A major problem of automating trade studies and the early phases of the design process is providing an environment which design engineers will use. Presently, they use pencil and paper to complete much of the design work in the early phases. Two features of this support are significant: (1) the ease with which they can capture and modify a design, and (2) the amount of design information they must provide to a tool before the tool provides them anything of value.

Design engineers sketch designs on paper without having to consider the mechanics of drawing. They do not need to remember how to get a tool to draw an instance of a particular part or a circuit, but can sketch a design without interrupting their flow of thought. To include a part in a design, the design engineer need only draw a rectangle and label it. Since the name of an item carries its properties, the design engineer need only erase one name and write in another to change a microprocessor to a gate array. To define a data path between one design element and another,

the design engineer must merely draw a line between the two and label it. To group a set of design elements into a higher level function or module, the design engineer need only draw a box around those involved. Since the properties are in the mind of the design engineer and elicited by the name (label) of an item, the properties change immediately when the design engineer changes the name.

This is not necessarily true later in the process when the design definition needs to be more exact and more properties are associated with an item. For instance, the representation of a microprocessor will include the number and function of pins. Changing a microprocessor to a gate array will likely change the number of pins and will certainly change their functions. The design engineer will need to update the connections between elements in the design appropriately to reflect this change. The level of detail the design engineer must supply to the ECAD system is consistent with the complexity of the task at this level. What is required is support for the actual activity at the preliminary design level and a seamless transition at the appropriate time to something like a current ECAD system.

Design engineers repeatedly emphasized these points to explain why they avoid using existing CAD tools for preliminary design tasks. The reason the tools are not used earlier in the design cycle seems to stem from the initial purpose of current CAD tools. They were meant to provide a way to aid design engineers in simulating a design. Since simulation models require detailed information with all the pins properly connected, the user interface reflects this requirement.

Although an attempt was made to address this problem in the prototyping effort, the design representation scheme employed by the NS ECAD workstation and the need to map entities into a design and parts library limited the options for building an interface. The problem was that the underlying representation had to support the proposed interface and information usage. Since Symbolics, Inc. developed the NS ECAD workstation primarily to support specific very-large-scale integration and circuit board design tasks, the design representation was not wholly adequate for unit and module design. Furthermore, its developers had not anticipated the tasks that were attempted under this effort. Therefore, it did not provide support for the style of interface discussed above and constrained the capabilities in the proof-of-concept tools.

One possible solution that should be explored is the use of a palette of figures, such as those currently found in most computer drawing programs. Selecting a figure from a continuously visible palette and dragging it somewhere on the screen should be no more disruptive than sketching a rectangle. The name of an item would determine its properties. Ideally, the system would recognize most domain names and associate appropriate properties with the object. However, these properties would be associated with the name instance rather than the object itself.

If the design engineer changed the name, the properties would change as appropriate to the new object as long as the system recognized the new name. In addition, the user should be able to associate arbitrary properties with any object. System developers should cooperate with several users to ensure the system facilitates the creative process by using an acceptably simple means of creating new properties and assigning values, adding component names to the set the system recognizes, and associating properties with the class representation of new or existing names.

Design Representation

Different representations of design information are required to efficiently perform design optimization analyses. Ideally, a suite of analysis tools that access a single data set of design information should be available to all levels of design engineers and specialists. They can use this information at different stages and levels of an evolving design. This suite of tools would provide meaningful information for high-level, incomplete designs as well as detailed design representations. However, the design representations in existing tools often do not provide the information needed to support the entire design process. They do not enable modeling of high-level design concepts within the same tool and at the same time as the modeling of lower-level detailed design definitions with the links necessary to use this information to support the proof-of-concept tool analyses. In those cases where the design representations could provide information throughout a good cross section of the design process, the information was not adaptable for use by a wide variety of specialists. Existing tools often require a level of completeness and detail that is not readily available to the design engineer either because the design is not sufficiently defined or additional analyses must be executed to obtain this information.

A potential problem with many of the existing tools that employ a single representation/description of the design is that the tool, unless the representation and the integrated analyses are robust, cannot give meaningful feedback until the design is completely defined. Few of the tools can work with information at a variety of completeness, abstraction, and quality levels. In addition, a tool that offers a single design representation that is appropriate for the entire design process was not found.

For example, design engineers can perform an early reliability analysis once the functions and types of components are known. An initial testability analysis, however, requires that the circuit topology be represented in addition to the information required for an initial reliability analysis. Although the design engineer can perform an improved reliability analysis if this additional information is used, the analysis cannot be performed as early in the design process. If the design representation does not support incomplete, high-level models of the design, the reliability analysis

can only be performed after the design engineer defines the topology — often much later in the design cycle.

VI. FUTURE RESEARCH

Many of the desired capabilities of a future RAMCAD-type design environment are presented in Section II of this report and discussed more thoroughly in a report by Kitzmiller & Anderson (1991). Each of these capabilities will not be described in this report. Some of these capabilities are merely straightforward applications of the current knowledge, experience, and existing technology. However, the items that require further research before implementation will be discussed. Because of the diversity and scope of the technologies that could dramatically affect the design process of electronic systems, especially in a RAMCAD design environment, it is difficult to “single out” specific areas needing research and to define an approach to accomplish that research. The extent of ongoing research and the speed of technology development create a problem when describing research requirements because they may quickly be overcome by events or more easily solvable by methods not mentioned here. Consequently, only general ideas on the possible approaches in those areas requiring future research are provided.

Many of the problems limiting the effectiveness and efficiency of the design process are not unique to the design of electronic systems or RM&S. Therefore, the research areas are divided into a general research area section, sections specific to each RM&S research area, and a section specifically devoted to the proof-of-concept tools created under this research effort.

General Research Areas

Research needs to be conducted in four general areas: (1) design, process, methodology, and knowledge; (2) design data and libraries; (3) design estimates and predictions; and (4) design monitoring and advice.

Design Process, Methodology, and Knowledge

In addition to developing design methods and techniques for disciplines such as testability and manufacturing, the effectiveness and efficiency of the overall design process and methodology must be improved. Although most design engineers believe they understand the basic design cycle and system engineering process, additional work is needed to understand the best way to integrate the analyses and data involved in each design discipline, tailor the basic cycle and process to particular design problems, and support the basic cycle and process with improved design methods

and computing aids. Clearly, there is much to learn about how to improve the design of complex systems (electronic, mechanical, or software) and how the design engineer may be best assisted in doing so.

Researchers must refine the understanding of design methods and techniques for specialty areas (e.g., design for reliability and DFT), and develop an integrated set of design rules and guidelines based on an improved understanding of how these specialty areas should interact. To do so, researchers need to develop and document more detailed information and process models of the current design processes.¹¹ This will provide a basis for determining how to improve the design process and identifying where improved design methods and computing technology will provide the greatest benefit. Researchers may then model alternative "improved" design processes as changes to a baseline model to facilitate an assessment of their relative merits.

Design Data and Libraries

Although much of the basic component information needed during the design process is available in handbooks, little is currently available on-line in a format or medium usable by the design engineers or design tools. To significantly improve the effectiveness of the design process, basic component information (e.g., component area, weight, power consumption, cost, and basic producibility information) and other design information (e.g., requirements, design guidelines, and standards) must be available in on-line databases accessible by the design engineers through the tools used in the design process.

Clearly, a major impediment to an efficient design process and the design of more reliable, maintainable, and supportable systems is the inability of many people working on the same design to conceptualize and work from a common model of the system being designed. The lack of common models severely limits the effectiveness of the interaction of the design disciplines. To resolve this problem, researchers must create some method to help system engineers define common models that each design discipline can understand and work from. A major barrier to developing common models is the difficulty design teams have in sharing design information and determining the core elements and relationships about which the teams need to make decisions. Thus, to remove this barrier, the model methods must be created and a suite of tools must be used so that all people working on a design use the same information set.

¹¹ The basic design cycle and system engineering process were not documented well enough for this research effort. Thus, this information was documented in a paper titled, "Electronic Design Process" (Kitzmiller & Anderson, 1991).

As part of the above problem, CAD/CAE software systems need to capture, manage, and organize key design information so that it is easily accessible to those involved in the design process. The information includes design and manufacturing information as well as mission and system requirements; functional decomposition and dependency information; allocations; predictions; results of key analyses; design attributes such as weight, area, and power, applicable design guidelines; and decisions made throughout the design cycle. The CAD/CAE software system should also capture the rationale for and the uncertainty of the allocations, predictions, analysis results, and decisions made during the design cycle.

Design Estimates and Predictions

Accurate estimates/predictions of item attributes (e.g., cost, area, power consumption, temperature, and reliability) are essential to the design process. In many cases, the basic data needed to compute these estimates are not readily available for the types of technologies and designs being considered. This is a major barrier to enabling effective trade studies prior to a detailed design and component selection. The design community needs a framework to integrate the diverse functional and physical models involved in the design process and to provide a means of tracking the uncertainty and confidence levels of each key design attribute. The lack of information on the quality of the analysis results complicates the interpretation and integration of the results obtained from numerous separate analyses. Current analyses rarely include error bounds or other confidence-related information. However, this data would greatly aid the design engineer during an evaluation of the results of an analysis. Additional work is needed to develop analyses capable of providing the engineer with information on the quality and significance of the analyses.

During the concept generation and preliminary design stages of the design process, predictions and allocations are based on comparisons between the system, subsystem, or line-replaceable module (LRM) being designed and similar systems already fielded. Until the design engineer selects actual hardware or designs the system, subsystem, or LRM in detail, the predictions and assessments depend on analyses of similar systems.

Design engineers often obtain data for such similarity analyses from military sources, such as the Navy 3M program and MODAS, or from the manufacturers. These sources list known subsystems from various military systems, along with their allocated, predicted, and field experience RM&S characteristics. Where there is no exact equivalent in function, packaging, or operational environment, the design engineers base their estimates on their best engineering judgment. For example, design engineers might base the initial MTBF for a proposed radar

system on the MTBF of a similar radar system already fielded, then adjust the MTBF requirement to reflect the design differences between the two systems.

At present, however, there is no defined and published method for performing a similarity analysis that is widely accepted. Consequently, the quality of the predicted parameters depends on the expertise of the engineers performing the analysis. This area should be researched to determine the extent to which standard approaches and methods can be developed for this analysis. Also, the research should include ways to develop computing aids to assist design engineers in performing this task. This research should entail:

- determining how designs can be characterized to support efficient access and retrieval;
- determining the rules and judgmental factors needed to equate a proposed design alternative with one in an experience database;
- determining the rules, algorithms, and judgmental factors needed to adjust the design data of existing designs to reflect the current design problem; and
- determining the degree of fidelity and data consistency needed in the experience database.

Similarly, developing a prototype tool would require developing a database of existing designs that contains the critical design data and developing a system that provides a similarity matching capability on the design characteristics of interest (e.g., number of LRUs, number of shop-replaceable units [SRUs], technology, testability factors, and operating environment) and the rules and judgmental factors needed to adjust the data.

Design Monitoring and Advice

Design engineers need an improved capability to monitor their designs for compliance with standard design methods and practices. Current experience indicates that in many instances design engineers disable or ignore existing design monitoring capabilities because they find them more distracting than useful. Additional research is needed to determine how best to monitor an evolving design to provide effective feedback to the engineer. In addition, any design-monitoring system should provide the engineer with effective feedback on the status of any goals (e.g., reliability, maintainability, and performance) the design engineer must meet.

Part of the design-monitoring process could include a system of design notes and guidelines for using specific types of components or technologies, or for incorporating specific design features or practices. This information must be readily accessible to the design engineer.

Currently, design engineers obtain this information from design guideline documents. To facilitate their use, notes and guidelines must be accessible from the engineer's workstation in an on-line, relational fashion. This will require, as a minimum, capturing the current notes and guidelines in a relational database structure and associating keywords with each guideline.

Reliability

Additional design research must be conducted in four major areas to improve the reliability of future designs: (1) reliability prediction and modeling, (2) failure modes analysis, (3) reliability allocation, and (4) reliability enhancement.

Reliability Prediction and Modeling

The most fundamental need in reliability design work is for the capability to predict the failure rate of an evolving design. Design engineers need reliability predictions of such metrics as MTBF and mean time between critical failure at each level in the design hierarchy and during each phase of the design process. They need these predictions to support design tasks such as:

- defining the system architecture and the minimum operational configuration,
- evaluating the reliability of design alternatives,
- establishing and allocating the reliability requirements and design targets,
- determining the level of fault tolerance needed, and
- verifying that reliability requirements have been met.

Besides automating the earlier use of piece part and stress reliability prediction methods, there is a need to support the further development of system reliability models. Although it is not feasible to develop a tool capable of fully automating all aspects of the reliability prediction and modeling process for an arbitrary system, particularly one employing sophisticated reconfiguration and redundancy schemes, it is possible to develop a tool or suite of tools capable of supporting a wide range of design problems. Difficulties arise in situations for which design engineers need complex Markov models and Monte Carlo simulations to aid in the evaluation of system readiness and performance. Basic research is needed in this area to develop the level of understanding necessary to determine the applicability and limitations of these techniques in the modeling and simulation of complex systems. This understanding is needed before CAD/CAE software designers can develop widely applicable tools that design engineers, who are not experts in the

techniques, can effectively and safely use. Some work in this area has already been performed and documented (Fleming et al., 1987; Goldfeld & Dubi, 1987; Sahner & Trivedi, 1987; and Babcock et al., 1987).

Failure Modes Analysis

Often the difference between a reliable system and an unreliable system can be attributed to the effective implementation in the former of mechanisms to detect and handle common or frequently recurring failure modes (e.g., loss of power and loss of data). To develop a coherent system-wide approach to failure management and testability, design engineers must consider the impact of the common failure modes on the system operation early in the design process.

However, there is a basic need for computer aids to assist in the modeling, tracking, and evaluation of failures and their impact on system operation. Research in this area should include developing methods and techniques that aid the design engineer in the following areas:

- developing failure and fault models at the appropriate levels of the design hierarchy,
- propagating the effects of common faults throughout the system,
- collating the effects of different failures into common failure modes, and
- evaluating approaches to monitor and contain failure modes.

Based on this research effort, creating design aids appears to be a feasible means of assisting the design engineer in performing selected failure modes analysis tasks. Basic research must be conducted to develop the level of understanding necessary to determine which aspects of this activity could be most usefully automated.

Reliability Allocation

In conjunction with early reliability prediction and modeling, the reliability allocation process must be automated. Reliability allocation refers to the process of allotting reliability requirements and targets to partitioned hardware resources. Reliability requirements that are achievable, consistent with system requirements such as the MCSP, and compatible with the criticality and purpose of the item being designed must be established at each level in the design hierarchy. Design engineers may need to incorporate redundant elements in some LRUs and SRUs to ensure they meet the allocated level of reliability. Budgets for these units must include adequate resources (e.g., area, power, and cooling) to ensure that design engineers can incorporate redundant

elements as needed. Thus, with proper prediction and modeling, the allocation process can help ensure reliability requirements are met.

To minimize the costs (in dollars, area, cooling, power, etc.) associated with achieving a level of reliability for an item, design engineers must derive the allocations in concert with similarity analyses and reliability predictions. The basic need in this area is to provide methods and models that design engineers can use to predict the resources required to achieve a desired level of reliability in an early design concept.

Reliability Enhancement

To efficiently design a system to be reliable, circuit and packaging design engineers must be able to assess and improve the reliability of a design with minimal assistance from a reliability specialist. This involves aiding the engineer in the following areas: (1) predicting the key reliability characteristics and overall cost (e.g., dollars, power consumption, and area), (2) identifying the major contributors to the failure rate of the design, and (3) identifying the areas and hardware changes that will yield the greatest reliability improvement for the least cost. To facilitate this process, design engineers need improved methods that help them identify and rank design changes according to their relative reliability improvement and cost without requiring the design engineer to actually modify the design and run comparative analyses. Part of the research required in this area was performed and the results incorporated into the SIDECAR tool. However, much more research is needed to fully utilize the concepts in this tool.

Maintainability

For electronic systems, the key maintainability design issues are associated with human factors (e.g., visibility, accessibility, and weight), preventive maintenance (e.g., calibration and clearing), and fault testing and isolation. Of these, fault testing and isolation are considered the most significant factors affecting system maintenance and support costs. The Integrated Test and Maintenance study, conducted by Boeing (Sahner & Trivedi, 1987) for the Air Force, concluded that fault testing and isolation account for approximately 35 percent of the organizational-level maintenance labor and 63 percent of the depot-level maintenance labor. This study also concluded that an integrated test and maintenance system could reduce labor and repair costs by 13 percent and LCCs by four percent.

As with reliability, additional research must be conducted to develop models that aid the design engineer in: (1) estimating the key maintainability characteristics of a proposed design,

(2) deriving optimal maintainability allocations, (3) identifying and ranking design changes that enhance the maintainability of the design, (4) developing accurate test and fault isolation capabilities, and (5) aiding the engineer in ensuring human factors issues associated with a design are adequately addressed.

Maintainability Prediction and Modeling

Design engineers need maintainability and testability predictions to support such tasks as establishing the maintenance concept, evaluating the maintainability of design alternatives, establishing and allocating maintainability design requirements and resources, determining the level of maintainability and fault tolerance needed to assure mission readiness, and enhancing the maintainability of a design. In addition to automating currently known methods of predicting the key maintainability design parameters, methods must be developed to predict other maintainability- and testability-related characteristics such as false alarm rates and the burden and effectiveness associated with various testability approaches.

As noted above, fault testing and isolation are currently the largest contributors to maintenance man-hours. The most critical prediction and modeling need is for methods to predict the cost and testability benefits of alternative designs and proposed testability and fault isolation techniques.

Maintainability Allocation

As with reliability, there is a need to provide aids to assist the engineer in deriving the system maintainability levels needed to achieve the design objectives and requirements, and in allocating these requirements to lower levels in the design hierarchy. For electronic systems, one key requirement is to support the process of determining and allocating testability-related requirements. Currently, this process is done with high-level specifications for the basic metrics for fault coverage and isolation and does not always consider the inherent capabilities of specific hardware types.

Basic research is needed in this area to help create methods and tools to aid the design engineer in predicting the maintainability characteristics of a design element, in predicting the testability burden associated with a given level of testability, and in conducting trade studies to determine how the maintainability requirements and resources should be allocated.

Maintainability Enhancement

Additional research is needed into the methods and techniques a design engineer can employ to quickly identify areas of a design that are deficient from maintainability and testability perspectives,

identify potential techniques to improve design maintainability and testability, and to select the technique or set of techniques that enable the design to most easily and cost-effectively meet the maintainability requirements. Along with the need for models to predict the maintainability characteristics of a design element and the testability burden required to achieve a level of testability, there is a need to develop a catalog of maintainability enhancement techniques and to determine which of these techniques provides the greatest improvement in maintainability for the least burden. This ability must also be incorporated into either an easy-to-use method or a computerized tool that will allow the design engineer to perform this function with little or no help from maintainability specialists.

Testability Design Aids

Electronics have seen increased use in the most critical functions of a design (e.g., the fly-by-wire F-16 aircraft); therefore, the need for testability has increased. Now, as the U.S. Air Force moves towards a two-tier maintenance concept, it is even more important to have the capability to unambiguously isolate faults to an LRM and verify the operational readiness of a system on the flightline. This, in turn, levies a requirement that each module be highly testable. Fault isolation and testing are currently the largest contributors to the number of man-hours needed for maintenance. The most widely used testability measures during design are percentage of faults detected, size of the fault isolation ambiguity group, and false alarm rate. Current methods and models can only approximate these measures. Design engineers need new methods or models to accurately predict each of these measures.

Researchers must perform basic research in the area of improved testability models to aid design engineers in designing for testability and performing testability tradeoffs. The improved models should aid the design engineer in such tasks as developing a testability strategy, estimating the type and quantity of test resources needed to achieve a level of testability, and determining how to best distribute test resources. The researchers must also develop a standard catalog (taxonomy) of design for testability techniques with associated decision criteria as part of the research.

Human Factors

Human factors can significantly impact system maintainability and time to repair. For avionics systems, mechanical design tools can best support this aspect of the design by providing three-dimensional computer modeling capabilities and by assisting the design engineer in complying with maintainability design practices and guidelines. While computerized man models are currently available and used in some design functions, design engineers normally use them to determine strength requirements, goodness of fit, visibility, and other ergonomic factors. There is

a need to assist design engineers in other areas as well (e.g., designing connectors that minimize the possibility of a technician bending one of the connecting pins or losing a retaining ring during maintenance without reducing the ease of the maintenance activity).

Supportability

The key supportability issues are standardization, modularity, commonality, and the future availability of the modules and components used in the design. Many support decisions are driven not just by these issues but by operational and logistic requirements and constraints such as mission, budget, and restrictions on the use of external test equipment.

Automated tools or improved methods are needed for the following three areas of supportability: (1) estimating the key supportability characteristics of electronic systems, (2) deriving optimal supportability allocations, and (3) identifying and ranking design changes that enhance the supportability of the design.

Key Supportability Characteristics Prediction and Modeling

The DoD requires most of its contractors to assess the LCC implications of competing design alternatives. Thus, contractors routinely use LSC and LCC models to estimate the cost impact of key design decisions. However, these models still require more research so that they can relate the major physical design characteristics (e.g., packaging technology) to maintenance and support costs at each level in the design hierarchy. These models must be integrated with system-level design tools and must be general enough for wide use. Logistics support engineers can then tailor or customize these models to a specific system design problem based on the system mission and resources.

In addition, basic research needs to be performed to create robust supportability prediction methods, techniques, and tools. These methods, techniques, and tools must support such tasks as establishing the support concept, evaluating the supportability of design alternatives, establishing and allocating supportability design requirements and resources, determining the level of R&M needed to assure mission readiness, and enhancing design supportability.

Supportability Allocation

There is a need for research into methods that can assist the design engineer in establishing design supportability requirements and allocating these requirements to lower levels in the design hierarchy. Cost is the key supportability attribute for which design engineers need help in

maintaining a constant estimation and monitoring capability. Unfortunately, design engineers are often forced to create designs that do not exceed a specific manufacturing cost instead of designing the system to minimize the total costs associated with owning the system (which usually requires more money up-front but much less downstream).

Supportability Enhancement

Design engineers must be provided effective feedback on the supportability implications of specific design alternatives. To help in this area, supportability models must be created that offer a means of estimating the supportability impacts of design alternatives (e.g., packaging choices, fault-tolerance approaches, and modularity and commonality concerns) and maintainability decisions (e.g., depot repair versus throwaway items).

Proof-of-Concept Tools

The results of the evaluation indicate that the proof-of-concept tools developed under this research effort could be extended as described below.

Future System for the Interactive Design and Computer Analysis of Reliability Research

SIDECAR does not, by itself, fulfill all the reliability analysis requirements during preliminary unit design. It does not, for example, provide a means of completely evaluating the impact of proposed design modifications or design alternatives on overall mission performance. To provide such a capability, a mission analysis tool similar to that of the Mission Reliability Model (MIREM) program needs to be integrated with the current SIDECAR analyses. Expanding SIDECAR so it can accept the outputs of several different forms of analysis tools and combine them to more fully explore the design space would be one area of future research and development.

Reliability Analysis. The general belief within Boeing is that design engineers should use MIL-HDBK-217E and other reliability prediction methods as one of several indicators of the reliability of a device. Design engineers should use them in conjunction with other indicators of unreliability, such as design rule and derating guideline violations, to help identify potential reliability problems and determine the relative reliability of design elements. There is a widely held belief, for example, within Boeing that the use of MIL-HDBK-217E or other failure rate predictions to determine the placement of components on a circuit board for temperature is an inappropriate use of this data. The approach recommended by the Boeing design community is to select components designed for the temperature environment in which they will operate. This

should include ensuring that there is a sufficient temperature "cushion" between the design temperature of the part and the temperature at which the design engineer expects the part to operate. In addition, the design engineer should work towards a uniform temperature distribution across the board during operation. This will minimize the number and temperature magnitude of "hot spots." Design rules normally used by design engineers include items such as not placing temperature-sensitive devices next to power dissipation devices unless they are upstream of the cooling flow. One possible enhancement to the SIDECAR program would be to develop such a qualitative analysis capability to augment the current analysis methods.

Design Exploration. Currently, the SIDECAR design exploration capability has been limited to evaluating the effect of changing the quality level or packaging of a component, or the effect of "swapping" abstract modules that provide the same function. A limitation of the current SIDECAR is that it uses a single predicate, the "function" property of an element, to locate a component or structure that the design engineer can substitute for the original element. While this approach is considered acceptable to some degree for component-level trades, it is considered inadequate for situations in which the design problem involves a more complicated structure. A possible future extension to SIDECAR would be to integrate it with a rule-based system tool to enable it to consider more complex design alternatives.

Although SIDECAR has the capability to ensure that a potential design change identified by the exploration routine will not violate simple user-defined constraints (e.g., maximum area, power, and parts cost), it does not have the capability to ensure it does not violate more complicated user-defined constraints (e.g., processor speed) or to use these constraints to direct the design space exploration. In conjunction with extending SIDECAR to handle more complex design constructs, SIDECAR should be enhanced to handle more complex constraints and to use these constraints to restrict the exploration routine.

Reliability Enhancement Techniques. At present, SIDECAR supports too few reliability enhancement techniques to effectively explore the design space associated with a design or design element. Additional reliability enhancement techniques must be implemented to ensure that the design space can be investigated for a wide range of design problems.

Future Statistical Testability Analyzer Research

Design Modification Suggestions. Currently, STA is only capable of identifying areas of inadequate fault and test coverage. It does not suggest design modifications to improve the testability nor suggest which areas of inadequate coverage the design engineer should address first. The tool does not, for example, indicate which areas of inadequate fault coverage the design

engineer could most readily address. STA could determine this data if it were altered to estimate the inherent testability of a circuit or the manner in which a particular test set tests a component. STA could also be augmented with heuristics and other design knowledge to more clearly identify testability problem areas and suggest possible design modifications.

Design Knowledge. A second area of research would be to explore how STA could aid the design engineer in recognizing synergy between techniques and structures used to enhance the fault tolerance of a system and those used to enhance its testability. For example, incorporating TMR into a design to improve its reliability (availability) provides structures that the design engineer can also use to improve the testability of the design. Ideally, design modification suggestions should take such synergistic cases into account as part of their recommendation.

Future Scan Assistant Research

Research and development of ScanIt capabilities could include two areas. First, incorporate information about long path lengths in scan regions and the failure probabilities of modules. This will further reduce the costs of test development and fault location. Second, provide estimates of the burdens associated with using boundary scan in terms of additional gate delays and extra hardware required. ScanIt should be able to determine this information from the set of scan registers and their effect on critical paths. The design engineer can then decide whether to change the current set of scan registers based on estimates for the cost of test development, fault location, performance, and the area for the current set of scan registers. For example, the system engineer can determine the subset of critical modules that should have boundary scan to facilitate on-line testing while the detailed designer can determine the placement of scan registers to isolate pieces of a large module that might otherwise be tested as a monolith.

Future Inherent Testability Analyzer Research

Many potential benefits and limitations of ITA require further research and evaluation. The applicability of ITA to more abstract and complex designs, its potential to support multiple-component test techniques, and its ability to deal more effectively with inconsistency and reconvergence should be the foci of future ITA-related research.

Complex and Abstract Parts and Designs. Future research needs to include modeling more complex parts (such as multipliers and microprocessors) to validate the ITA concept and evaluate its potential usefulness. Modeling such abstract devices as generic microprocessors and memories will help facilitate the use of ITA earlier in the design cycle.

In the earliest phases of the design cycle, design engineers use abstract modules such as "central computer" or "engine control unit" to represent the design. They need methods for selecting and estimating the costs and benefits of various test alternatives based on the available data for such abstract modules. Further research is needed to refine and implement the methods for dealing with these abstract modules.

Multicomponent Test Techniques. ITA was designed around an isolated-component test paradigm in which individual components are tested by providing the necessary stimuli, observing the outputs, and comparing them to the required outputs. ITA does not support other test techniques involving multiple components including replication and comparison (e.g., modular redundancy) and consistency checks (e.g., parity). Methods other than controllability and observability propagation for recognizing and evaluating test techniques that involve components in fault-detection roles need to be developed.

Inconsistency and Reconvergence. Ignoring inconsistency- and reconvergence-related problems allows tools such as ITA to run much faster, be less sensitive to design ambiguities, and require less complex models. A drawback to this approach is that it reduces the class of testability problems that the tool can identify. Research to evaluate the magnitude of this drawback, particularly with abstract designs, has not been conducted. It is expected that the benefits will outweigh the costs in many areas of design, making ITA a worthwhile tool to use for a large percentage of design problems.

Problem Analysis. ITA identifies components that, because of several classes of problems, cannot be tested. The current version does not explicitly identify the causes of the problems or suggest solutions. It does, however, provide information on the controllability and observability of both the untestable component pins and of other network nodes. The design engineer can use this information to solve the problems. It is also possible to automate some or all of this process by having ITA identify controllability and observability test needs at the pins of untestable components. ITA, using its component test models, can determine the additional controllability and observability needed to test each untestable component. Several solutions resulting in varying amounts of testability improvement may be possible. By analyzing the test needs (from the preceding step) of neighboring components, some problems solvable by common solutions might be apparent. The design engineer might cure a large problem involving a string of components that have uncontrollable inputs with a single fix. Placing advanced heuristics in an intelligent ITA design adviser so that it performs functions such as adding controllability to the front of a string of problem components (such as those described in the previous sentence) could simplify many of the problems design engineers constantly face.

ITA could also propose that controllability and/or observability be added at various points by integrating it with a tool such as ScanIt.

Unneeded Controllability and Observability Features. Design engineers could also use a modified version of ITA to identify unneeded test hardware. They could use the modified ITA to find possible redundancies in the controllability and observability of a piece of hardware. However, ITA would probably only be able to identify possible redundancies, not conclusively prove that resources are redundant. For example, a design engineer might add extra hardware to resolve a consistency problem of which ITA would be unaware.

Several methods of implementing this are possible. One would involve expanding the P-algebra to indicate whether controllability and observability are derived from test hardware. Another would involve set operations (such as complements and intersections) on analyses performed with and without the possibly redundant controllability and observability contributions of resources.

Integrated Proof-of-Concept Tools

Currently, STA is able to automatically execute SIDECAR to obtain reliability information for the testability analysis it performs. Additional work is needed to consolidate the SIDECAR and STA tools (the SIDECAR and STA data structures, for instance, could be integrated) and to ensure that the SIDECAR and STA analyses provide consistent results. The exploration routine could be modified to consider the testability of proposed design changes as well as their reliability impact.

Another area in which integrating these tools would be helpful is ITA and STA. ITA could be used to provide much of the testability information STA requires. Similarly, the ScanIt program could be integrated to provide the design engineer feedback on where additional control and observation points are needed and where scan could be cost-effectively incorporated to improve the inherent testability of a design.

REFERENCES

- Babcock, P., Leong, F., & Gai, E. (1987). *On the next generation of reliability analysis tools* (NASA Contractor Report No. 178380). Unpublished manuscript.
- Bodker, S., Ehn, P., Knudsen, J.L., Kyng, M., & Madsen, K. (1988). Computer support for cooperative design. In *Proceedings of the Conference on Computer-Supported Cooperative Work*. New York: Association for Computing Machinery.
- Boeing Computer Services. (1988). *RAMCAD interim report on research and development: Analyze current CAD capabilities*. Unpublished manuscript, Seattle, WA.
- Boeing Computer Services. (1989). *RAMCAD interim report on research and development: RM&S areas to be addressed*. Unpublished manuscript, Seattle, WA.
- Boeing Computer Services. (1990a). *RAMCAD software development program research plan*. Unpublished manuscript, Seattle, WA.
- Boeing Computer Services. (1990b). *RAMCAD interim report on research and development: RM&S design techniques*. Unpublished manuscript, Seattle, WA.
- CALS Summer Study. (1988). *Integration of R&M into the automated design process* (Report of the CALS R&M Summer Study on Complex Electronics). Unpublished manuscript.
- Department of Defense, Military Handbook 217E, Reliability Prediction of Electronic Equipment, January 1982.
- Department of Defense, Military Standard 472, Maintainability Prediction, 1966.
- Department of Defense, Military Standard 785B, Reliability Prediction for Systems and Equipment Development and Production, 1986.
- Department of Defense, Military Standard 1388-1A, Logistic Support Analysis, April 1983.
- Department of Defense, Military Standard 1388-2B, DoD Requirements for a Logistics Support Analysis Record, 1991.
- Department of Defense, Military Standard 2165, Testability Program for Electronic Systems and Equipments, 1985.
- Department of Defense, Military Standard 1629A, Procedures for Performing a Failure Mode Effects and Criticality Analysis, November 1980.
- Fleming, R., Josselyn J., & Boyle P. (1987). Integrated design of modular avionics for performance and supportability. In *Proceedings of the National Aerospace Electronics Conference*. New York: Institute of Electrical and Electronics Engineers.
- Goldfeld, A., & Dubi, A. (1987, September). Monte Carlo methods on the computation of system reliability. *Energia Nucleare*.
- Kitzmiller, C.T., & Anderson, M.A. (1991). *Electronic Design Process* (AFHRL-TP-90-34, AD-A234 709). Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.
- Kitzmiller, C.T., Anderson, M.A., & Tracy, M.C. (1993). *RAMCAD Design Methodology* (In press). Wright-Patterson AFB, OH: Logistics Research Division, Armstrong Laboratory.
- Primrose, P.L. (1988). Selecting and evaluating cost-effective CAD/CAM. *Computer-Aided Engineering Journal*, 5(1), 33-38.
- Sanner, R., & Trivedi, K. (1987). Reliability modeling using SHARPE. *IEEE Transactions on Reliability* VR-36(2), 186-193.

- Tracy, M.C. (1991). Implementing MPT design requirements through RAMCAD. In *Human-Centered Technology for Maintainability: Workshop Proceedings* (AL-TP-1991-0010, AD-A239 090). Wright-Patterson AFB, OH: Logistics Research Division, Armstrong Laboratory.
- Yount, C.R., & Siewiorek, D.P. (1991). Sidecar: Design support for reliability. In *1991 Design Automation Conference Proceedings*. New York, NY: American Society of Mechanical Engineers.

BIBLIOGRAPHY

- Alexander, A.J. (1988). *The Cost and Benefit of Reliability in Military Equipment* (Rand Report P-7515). Santa Monica, CA: The RAND Corporation.
- Azam, S., Naft, J., Pecht, M., & Richter, K.J. (1988). *Decision Support Requirements in a Unified Life Cycle Engineering (ULCE) Environment Volume II: Conceptual Approaches to Optimization* (IDA Paper P-2064). Alexandria, VA: Institute for Defense Analyses.
- Boeing Computer Services. (1989). *RAMCAD interim report on research and development: Evaluation criteria*. Unpublished manuscript, Seattle, WA.
- Boeing Computer Services. (1990). *RAMCAD interim report on research and development: Evaluation methods*. Unpublished manuscript, Seattle, WA.
- Boeing Computer Services. (1990). *RAMCAD interim report on research and development: Description of software tool*. Unpublished manuscript, Seattle, WA.
- Boeing Computer Services. (1990). *RAMCAD interim report on research and development: Demonstration and evaluation of the optimization software tool*. Unpublished manuscript, Seattle, WA.
- Bhandari, I., Beroul, P., & Siewiorek, D. (1990). *ScanIt: An experiment in system-level testability*. Unpublished manuscript, Pittsburgh, PA.
- Cralley, W.E., Dierolf, D., & Richter, K.J. (1990). *Computer Support for Conducting Supportability Trade-Offs in a Team Setting* (IDA Paper P-2313). Alexandria, VA: Institute for Defense Analyses.
- Defense Systems Management College. (1990). *System engineering management guide*. Fort Belvoir, VA: Defense Systems Management College.
- Edmond, P. (1989). *Automated synthesis of reliable systems*. Unpublished master's thesis, Carnegie-Mellon University, Pittsburgh, PA.
- Elkind, S.A. (1982). Reliability and availability techniques. In D.P. Siewiorek & R.S. Swarz (Eds.), *The Theory and Practice of Reliable System Design* (Chapter 3). Bedford, MA: Digital Press.
- Fulton, R.E., Pin-Yeh, C., & Richter, K.J. (1989). *Managing Engineering Design Information* (IDA Paper P-2154). Alexandria, VA: Institute for Defense Analyses.
- General Dynamics, Convair Division. (1988). *RAMCAD software survey final report*. Unpublished manuscript, San Diego, CA.
- Gebman, J.R., McIver, D.W., & Shulman, H.L. (1989). *A New View of Weapon System Reliability and Maintainability* (Rand Report R-3604/1-AF, R-3604/2-AF). Santa Monica, CA: The RAND Corporation.
- Goldstein, L.H. (1979). Controllability/observability analysis of digital circuits. *IEEE Transactions on Circuits and Systems*, CAS-26(9), 685-693.
- Landers, T.L., Taha, H.A., & King, C.L. (1990). *A reliability simulation approach for design engineers*. Unpublished manuscript.
- Leonard, C.T., & Pecht, M. (1990). How failure prediction methodology affects electronic equipment design. *Quality and Reliability Engineering International*, 6(4).
- Park, G., & Arora, J.S. (1986). *Interactive Design Optimization with Modern Database Management Concepts* (AFOSR-TR-87-0250, AD-A177337). Washington, DC: Air Force Office of Scientific Research.

- Siewiorek, D.P., & Swarz, R.S. (1982). *The theory and practice of reliable system design*. Bedford, MA: Digital Press.
- Silberman, G.M., & Spillinger, I. (1990). Using functional fault simulation and the difference fault model to estimate implementation fault coverage. *IEEE Transactions on Computer-Aided Design*, 9(12), 1335-1343.
- Stephenson, J., & Grason, J. (1976). *A testability measure for register transfer level digital circuits*. Paper presented at the 6th International Symposium of Fault-Tolerant Computing.
- TRW. (1988). *RAMCAD interim report: Software identification and evaluation*. Unpublished manuscript.
- Ullman, D.G., Dietterich, T.G., & Stauffer, L.A. (1988). A model of the mechanical design process based on empirical data. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2(1), 33-52.
- Van Den Dam, R.F. (1985). *A perspective of mathematical simulation and optimization techniques in computer aided design*. Paper presented at the 2nd International Conference on Computer Applications in Production and Engineering (CAPE 86), Copenhagen, Denmark.
- Veatch, M., & Gates, R. (1986). *Mission Reliability Model Users Guide* (AFHRL-TR-86-35, AD-A175 235). Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.
- Yager, R. (1978). *Competitiveness and Compensation in Decision Making: A Fuzzy Set Base Interpretation* (Iona College Technical Report RRY 78-14). New Rochelle, NY: Iona College.
- Zeleny, M. (1982). *Multiple criteria decision making*. New York: McGraw-Hill Book Company.

ABBREVIATIONS

AI	artificial intelligence
AL/HRG	Armstrong Laboratory, Logistics Research Division
ASSURE	Automated Synthesis of Reliable Systems
BCS	Boeing Computer Services
BITE	built-in test equipment
CAD/CAE	computer-aided design/computer-aided engineering
CALS	Computer-Aided Acquisition Logistics Support
CMU	Carnegie-Mellon University
DFT	design for testability
DoD	Department of Defense
ECAD/ECAE	electronic computer-aided design/electronic computer-aided engineering
IPD	integrated product development
ITA	Inherent Testability Analyzer
LCC	life-cycle cost
LRM	line-replaceable module
LRU	line-replaceable unit
LSC	logistics support cost
MCSP	mission completion success probability
MICON	Microprocessor Configurer
MIL-HDBK	military handbook
MIL-STD	military standard
MIREM	Mission Reliability Model
MODAS	Maintenance On-Line Data Acquisition System
MTBF	mean time between failure
MTTF	mean time to failure
MTTR	mean time to repair
Pfd	probability of fault detection
R&M	reliability and maintainability
RAMCAD	Reliability, Availability, and Maintainability in Computer-Aided Design
RH&G	rule, heuristic, and guideline
RM&S	reliability, maintainability, and supportability

ScanIt	Scan Assistant
SIDECAR	System for the Interactive Design and Computer Analysis of Reliability
SRU	shop-replaceable unit
STA	Statistical Testability Analyzer
TMR	triple-modular redundancy